



TESIS - TE142599

**INTEROPERABILITAS TINGKAT FUNGSIONALITAS APLIKASI
PADA MIGRASI *VIRTUAL MACHINE* DI LINGKUNGAN *CLOUD
COMPUTING***

SOFFA ZAHARA
2214206202

DOSEN PEMBIMBING
Dr. Surya Sumpeno S.T., M.Sc.
Dr. Istas Pratomo S.T., M.T.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TESIS - TE142599

**INTEROPERABILITAS TINGKAT FUNGSIONALITAS APLIKASI
PADA MIGRASI VIRTUAL MACHINE DI LINGKUNGAN CLOUD
COMPUTING**

**SOFFA ZAHARA
2214206202**

**DOSEN PEMBIMBING
Dr. Surya Sumpeno S.T., M.Sc.
Dr. Istas Pratomo S.T., M.T.**

**PROGRAM MAGISTER
BIDANG KEAHLIAN TELEMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

LEMBAR PENGESAHAN

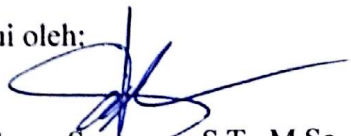
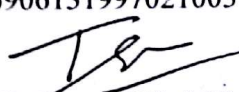
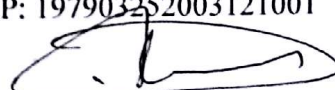

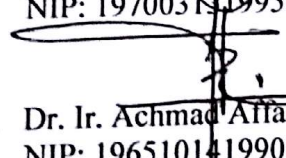
Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Soffa Zahara
NRP. 2214206202

Tanggal Ujian : 6 Januari 2017
Periode Wisuda: Maret 2017

Disetujui oleh;

- 
1. Dr. Surya Sumpeno, S.T., M.Sc. (Pembimbing I)
NIP: 196906131997021003
- 
2. Dr. Istas Pratomo, ST., MT. (Pembimbing II)
NIP: 197903252003121001
- 
3. Mochamad Hariadi, ST., M.Sc., Ph.D. (Penguji)
NIP: 196912091997031002
- 
4. Dr. Supeno Mard Susiki Nugroho, ST., MT (Penguji)
NIP: 197003131995121001
- 
5. Dr. Ir. Achmad Affandi, DEA (Penguji)
NIP: 196510141990021001

an. Direktur Program Pascasarjana
Asisten Direktur



Prof. Dr. Ir. Tri Widjaja, M.Eng.
NIP. 19611021 198503 1 001

Direktur Program Pascasarjana,

Prof. Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 196012021987011001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “**INTEROPERABILITAS TINGKAT FUNGSIONALITAS APLIKASI PADA VIRTUAL MACHINE DI LINGKUNGAN CLOUD COMPUTING**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 12 Januari 2017

Soffa Zahara
NRP. 2214206202

Halaman ini sengaja dikosongkan

INTEROPERABILITAS TINGKAT FUNGSIONALITAS APLIKASI PADA VIRTUAL MACHINE DI LINGKUNGAN CLOUD COMPUTING

Nama mahasiswa : Soffa Zahara
NRP : 2214206202
Pembimbing : 1. Dr. Surya Sumpeno, S.T., M.Sc.
2. Dr. Istas Pratomo, S.T., M.T.

ABSTRAK

Pesatnya perkembangan internet diiringi dengan meningkatnya aplikasi yang memanfaatkan teknologi *cloud computing*. Bertambahnya proses bisnis suatu aplikasi dalam suatu sistem juga fitur layanan yang ditawarkan *cloud provider* lain yang lebih baik, seperti harga yang lebih murah, performansi sistem yang lebih tinggi, sistem yang lebih handal, teknologi *security* terbaru yang digunakan serta SLA(Service Level Agreement) yang lebih baik, memungkinkan untuk berpindah layanan ke *cloud provider* lain. Namun, dalam prakteknya sering terjadi kegagalan fungsionalitas aplikasi dalam proses migrasi ke sistem *cloud* yang baru yang salah satunya disebabkan oleh masalah *vendor lock-in*. *Vendor lock-in* muncul disebabkan banyaknya *cloud provider* yang mempunyai standar tersendiri dalam pembangunan sistemnya yang mengakibatkan terbatasnya kemampuan migrasi antar *cloud*.

Penelitian ini memperkenalkan metode baru untuk menguji interoperabilitas sistem khususnya di tingkat fungsionalitas aplikasi diantara dua *cloud provider* IaaS (Infrastructure as a Service). Tujuannya adalah untuk mengetahui tingkat interoperabilitas aplikasi dalam sistem *virtual machine* yang berpindah dari sebuah cloud provider ke *cloud provider* lain. Pengujian migrasi yang dilakukan dalam penelitian ini terbagi menjadi 2 skenario yaitu dari Amazon EC2 ke Indonesian Cloud, kemudian dari Amazon EC2 ke Google Compute Engine. Dari hasil pengujian yang didapatkan bahwa tingkat interoperabilitas migrasi dari Amazon EC2 ke Google Compute Engine lebih tinggi dikarenakan pada migrasi Amazon EC2 ke Indonesian Cloud terdapat fungsionalitas aplikasi yang tidak berjalan.

Kata kunci: Interoperabilitas, Virtual Machine, Migrasi Cloud, IaaS, Aplikasi

Halaman ini sengaja dikosongkan

APPLICATION LEVEL INTEROPERABILITY ON VIRTUAL MACHINE IN CLOUD COMPUTING ENVIRONMENT

By : Soffa Zahara
Student Identity Number : 2214206202
Supervisor(s) : 1. Dr. Surya Sumpeno, S.T., M.Sc.
2. Dr. Istas Pratomo, S.T., M.T.

ABSTRACT

The rapid development of internet nowadays impact on increasing many applications that utilizing cloud computing technology for the sake of organization. Increasing requirements in applications caused by inevitable business process growth in organization, increased security, better availability and QOS enabling a tendency to switch from old cloud provider to more reliable one. However, in practice, application functionality failures often occur in case of migrating process to new cloud system due to several circumstances e.g. vendor lock-in problem.

This paper introduces a development method for system migration testing between two cloud providers. The goal is to determine the interoperability level of application in virtual machine within hypervisor system that moves from one cloud provider to another cloud provider. Migration testing is done by 2 types of scenario. First, from Amazon EC2 to Indoensian Cloud. Second, from Amazon EC2 to Google Compute Engine. After several testings, the result is the interoperability level of migration from Amazon EC2 to Google Compute Engine is higher than Amazon EC2 to Indonesian Cloud caused by there is application functionality that can not work after migration.

Key words: Interoperability, Virtual Machine, Cloud Migration, IaaS, Application

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puja dan puji syukur kepada Allah SWT atas segala rahmat dan karunia yang telah dilimpahkan kepada penulis sehingga penulisan tesis dengan judul :

“APPLICATION LEVEL INTEROPERABILITY ON VIRTUAL MACHINE IN CLOUD COMPUTING ENVIRONMENT”

Dapat diselesaikan dengan baik. Buku tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar magister pada program studi Teknik Elektro dengan bidang keahlian Telematika, Institut Teknologi Sepuluh Nopember.

Pada kesempatan ini penulis sampaikan terima kasih yang sedalam-dalamnya kepada :

1. Kedua orangtua, suami, adik, serta keluarga yang selalu memberikan dorongan semangat dalam mengerjakan dan menyelesaikan tesis ini.
2. Bapak Surya Sumpeno dan Istas Pratomo atas bimbingan, kesabaran dan pendorong semangat dalam menyelesaikan thesis ini.
3. Rekan-rekan lab Jaringan Telekomunikasi B301, rekan-rekan jurusan telematika, terima kasih atas kebaikan dan kerjasamanya dalam penelitian ini.

Penulis menyadari bahwa dalam penulisan tesis ini masih jauh dari kata sempurna, untuk itu demi perbaikan dan penyempurnaan tesis ini maka saran dan kritik membangun sangat diharapkan. Besar harapan penulis bahwa buku tesis ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa jurusan Teknik Elektro pada khususnya.

Surabaya, 11 Januari 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Kontribusi	3
1.6 Metodologi Penelitian	3
BAB 2 KAJIAN PUSTAKA	7
2.1 Cloud Computing	7
2.1.1 Amazon EC2	11
2.1.2 Indonesian Cloud	13
2.1.3 Google Compute Engine	14
2.2 Teknologi Virtualisasi	15
2.3 Migrasi Cloud	20
2.4 Interoperabilitas	22
2.5 Pengujian Aplikasi	25
2.6 Penelitian Sebelumnya	30
BAB 3 METODE PENELITIAN	35
3.1 Metode Penelitian	35
3.2 Rancangan Sistem	36
3.2.1 Skenario Desain Migrasi Sistem	37

3.2.2	Aplikasi E-Commerce	38
3.3	Skenario Pengujian.....	53
3.4	Rancangan Evaluasi Hasil Pengujian	54
3.5	Rancangan Parameter Level Evaluasi Interoperabilitas	58
BAB 4 HASIL DAN PEMBAHASAN		61
4.1	Migrasi Amazon EC2 ke Indonesian Cloud.....	61
4.2	Migrasi Amazon EC2 ke Google Compute Engine	71
BAB 5 KESIMPULAN DAN SARAN		83
5.1	Kesimpulan.....	83
5.2	Saran.....	84
DAFTAR PUSTAKA.....		85
LAMPIRAN		89

DAFTAR GAMBAR

Gambar 1.1 Metodologi Penelitian	3
Gambar 2.1 Model Cloud Computing NIST (Sosinky, 2011)	8
Gambar 2.2 Amazon EC2 Hypervisor (Badola, 2015)	12
Gambar 2.3 Fitur Amazon VM Export/Import (Amazon Web Services, 2016)...	13
Gambar 2.4 Arsitektur Hypervisor VMWare (Amazon Web Services, 2016)	14
Gambar 2.5 Arsitektur layanan Google Compute Engine	15
Gambar 2.6 Arsitektur Hypervisor Google Cloud Engine.....	15
Gambar 2.7 Prinsip Virtualisasi	16
Gambar 2.8 Perbedaan 2 Jenis Hypervisor (Sosinky, 2011).....	18
Gambar 2.9 Model McCall (Galin, 2004).....	26
Gambar 2.10 Proses Pengujian (Galin. D, 2004).....	27
Gambar 2.11 Metode TIOSA (Lenk dkk, 2014).....	30
Gambar 2.12 Hasil Pengujian CentOS 6.6 (Lenk dkk, 2014).....	31
Gambar 2.13 Hasil Pengujian Ubuntu 12.04 (Lenk dkk, 2014).....	32
Gambar 2.14 Hasil Pengujian Windows Server 2008 R2 (Lenk dkk, 2014)	32
Gambar 2.15 Hasil Keseluruhan (Lenk dkk, 2014)	33
Gambar 3.1 Metode Penelitian.....	35
Gambar 3.2 Skenario Migrasi Pertama	37
Gambar 3.3 Skenario Migrasi Kedua.....	38
Gambar 3.4 Diagram Use Case Aplikasi Pengujian	39
Gambar 3.5 Diagram Aktivitas Mesanan Barang	41
Gambar 3.6 Diagram Aktivitas Menambah Komentar Pada Artikel.....	42
Gambar 3.7 Diagram Aktivitas Menambah Produk	43
Gambar 3.8 Diagram Aktivitas Mengubah Produk	44
Gambar 3.9 Diagram Aktivitas Menghapus Produk.....	45
Gambar 3.10 Diagram Aktivitas Menambah Review Produk	46
Gambar 3.11 Diagram Aktivitas Menambah Artikel Baru.....	47
Gambar 3.12 Diagram Aktivitas Mengubah Artikel	48
Gambar 3.13 Diagram Aktivitas Menghapus Artikel.....	49
Gambar 3.14 Diagram Aktivitas Report	50
Gambar 3.15 Diagram Aktivitas Menambah User Baru.....	51
Gambar 3.16 Diagram Aktivitas Mengubah User	52
Gambar 3.17 Diagram Aktivitas Menghapus User.....	53
Gambar 4.1 Konfigurasi Hardware Server Fisik.....	67
Gambar 4.2 Konfigurasi Hardware Server Amazon EC2.....	67
Gambar 4.3 Perubahan Konfigurasi Aplikasi	68
Gambar 4.4 Akses Aplikasi Setelah Migrasi Dari Server Fisik.....	68
Gambar 4.5 Proses Perubahan Image VM dari Server Fisik ke Server Indonesian Cloud.....	69

Gambar 4.6 Gambar Banner Website Tidak Muncul Setelah Migrasi.....	71
Gambar 4.7 Proses Booting Berhasil di Server Google Compute Engine Setelah Migrasi.....	75
Gambar 4.8 Proses Perubahan Image VM dari Amazon EC2 ke Google Compute Engine.....	76

DAFTAR TABEL

Tabel 2.1 Metodologi Cloud-RMM (Pappas, 2014)	21
Tabel 2.2 Metodologi ARTIST(Pappas, 2014)	22
Tabel 3.1 Rancangan Testbed Fungsionalitas Aplikasi	54
Tabel 3.2 Rancangan Testbed Pengujian Migrasi VM	57
Tabel 3.3 Deskripsi Level Interoperabilitas Aplikasi	59
Tabel 3.4 Evaluasi Hasil Keseluruhan	60
Tabel 4.1 Hasil Pengujian Migrasi Image VM Amazon EC2 ke Indonesian Cloud	64
Tabel 4.2 Hasil Pengujian Fungsionalitas Aplikasi dari Amazon EC2 ke Indonesian Cloud	64
Tabel 4.3 Hasil Pengujian Migrasi Image VM Amazon EC2 ke Google Compute Engine	74
Tabel 4.4 Hasil Pengujian Fungsionalitas Aplikasi dari Amazon EC2 ke Google Compute Engine	77
Tabel 4.5 Hasil Evaluasi Secara Keseluruhan.....	78
Tabel 4.6 Perbandingan Hasil dengan Penelitian Sebelumnya.....	79
Tabel 4.7 Perbedaan Teknologi Migrasi Antar Provider	80

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Beragamnya *cloud provider* atau penyedia layanan *cloud computing* mengakibatkan munculnya masalah *vendor lock-in*. *Vendor lock-in* merupakan keadaan dimana pengguna suatu produk bergantung pada satu *provider* dan tidak dapat berpindah ke *provider* lain tanpa biaya yang besar dan waktu yang lama (Opera-Martins dkk, 2014). Setiap *cloud provider* memiliki format tersendiri dalam membangun produk layanan yang dimiliki yang menyebabkan terbatasnya kemampuan sistem bermigrasi antar *cloud*. Dalam proses migrasi atau perpindahan dari satu sistem ke sistem lainnya terutama dalam lingkungan *cloud computing* kemampuan interoperabilitas menjadi faktor yang penting. Interoperabilitas *cloud* merupakan kemudahan migrasi dan integrasi aplikasi dan data diantara penyedia layanan *cloud computing* (Dowell dkk, 2011).

Virtual Machine (VM) merupakan bagian terkecil dalam sistem khususnya dalam layanan IaaS (Infrastructure as a Service) yang dapat dilakukan proses migrasi (Lenk dkk, 2014). Dalam proses migrasi IaaS, berhasilnya proses konversi *image virtual machine* dari satu sistem ke sistem lain tidak menjamin berhasilnya proses migrasi. Terdapat kecenderungan beberapa parameter seperti konfigurasi jaringan, CPU, memory, parameter security, struktur aplikasi dan data yang mungkin berubah dalam proses migrasi. Masalah lain yang dapat terjadi dalam proses migrasi di level IaaS yaitu ketika aplikasi yang berjalan telah terintegrasi secara kompleks di sistem lama sehingga sistem sulit melakukan migrasi di sistem baru (Villary dkk, 2012).

Dalam proses migrasi sistem IaaS yang di dalamnya terdapat beberapa aplikasi yang telah berjalan seperti aplikasi *website*, dari suatu *cloud provider* ke *cloud provider* lain dibutuhkan suatu metode pengujian untuk mengukur tingkat interoperabilitas dari kedua sistem tersebut sehingga seluruh komponen dalam sistem seperti sistem operasi, sistem keamanan, aplikasi dan data yang tersimpan, dapat berjalan sesuai dengan fungsionalitasnya di sistem baru.

TIOSA (Lenk dkk, 2014) merupakan suatu metode pengujian interoperabilitas pada Virtual Machine berbasis Open Data Center Alliance(ODCA). Kekurangan dari metode TIOSA ini yaitu hanya menguji dari sisi fungsionalitas sistem operasi dan *hypervisor* dan belum menguji pada tingkat fungsionalitas yang lebih spesifik yaitu aplikasi di sistem yang baru. Selain itu pengujian interoperabilitas yang dilakukan masih dalam tahap simulasi di jaringan lokal sehingga belum mendekati evaluasi interoperabilitas di sistem sebenarnya khususnya di bidang industri.

Tujuan dari penelitian ini yaitu untuk memperkenalkan metode untuk mengukur tingkat interoperabilitas fungsionalitas aplikasi yang bermigrasi diantara dua *provider cloud* IaaS yang berbeda. Manfaat dari penelitian ini yaitu memberikan referensi metode pengujian interoperabilitas migrasi sistem antar *cloud* yang lebih spesifik ke fungsionalitas aplikasi sehingga dapat diimplementasikan di bidang industri sesuai proses bisnis yang berjalan.

1.2 Rumusan Masalah

Munculnya masalah *vendor lock-in* yang disebabkan banyaknya *cloud provider* yang mempunyai format dan tipe pembangunan sistem yang berbeda membatasi kemungkinan migrasi aplikasi diantara *cloud provider* tersebut. Selain itu, belum tersedianya metode yang menguji tingkat interoperabilitas sistem yang bermigrasi diantara dua lingkungan cloud IaaS pada tingkat fungsionalitas aplikasi menyebabkan banyak organisasi kesulitan untuk mengetahui tingkat interoperabilitas sistem *cloud* tujuan ketika bermigrasi.

1.3 Tujuan

Tujuan utama dari penelitian ini yaitu memperkenalkan metode untuk menguji tingkat interoperabilitas aplikasi pada sistem yang bermigrasi pada dua *cloud* IaaS yang berbeda sehingga dapat dijadikan referensi untuk menghindari masalah *vendor lock-in* yang sering terjadi di lingkungan *cloud computing*.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini yaitu :

1. Model layanan *cloud* yang digunakan yaitu IaaS(Infrastructure as a Service)
2. Aplikasi yang akan diuji menggunakan E-Commerce
3. Penelitian yang dilakukan tidak membahas interoperabilitas data dalam sistem
4. Penelitian yang dilakukan tidak membahas dari sisi keamanan sistem
5. Jenis *cloud* provider yang akan digunakan yaitu Amazon EC2, Indonesian Cloud dan Google Compute Engine
6. Sistem operasi yang akan digunakan yaitu Ubuntu 12.04

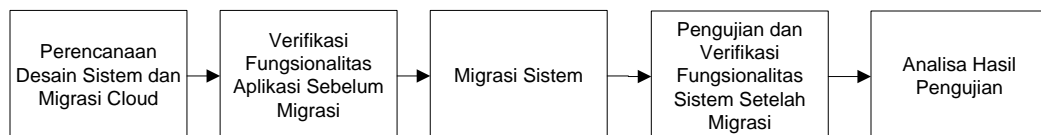
1.5 Kontribusi

Kontribusi dari penelitian ini yaitu:

1. Memberikan referensi metode pengujian interoperabilitas migrasi sistem antar *cloud* yang lebih spesifik ke aplikasi sehingga dapat diimplementasikan di bidang industri sesuai proses bisnis yang berjalan.
2. Memberikan referensi untuk menghindari masalah *vendor lock-in* dalam proses migrasi antar *cloud provider* terutama dalam layanan IaaS.
3. Memberikan kontribusi pengembangan metode pengujian interoperabilitas pada *cloud computing* khususnya pada jenis layanan IaaS.

1.6 Metodologi Penelitian

Tahapan metodologi penelitian yang akan dilakukan pada penelitian ini yaitu:



Gambar 1.1 Metodologi Penelitian

Tahapan – tahapan yang dilalui dalam penelitian ini adalah sebagai berikut:

1. Perencanaan Desain Sistem dan Migrasi Cloud

Dalam tahapan perencanaan desain sistem *cloud*, akan dilakukan desain skenario sistem *cloud* yang akan dibangun dan diuji baik dari *cloud* sumber maupun *cloud* tujuan termasuk pemilihan jenis *cloud* dan *cloud provider* yang akan dipilih.

Dalam tahap perencanaan migrasi terdapat beberapa proses yang dilakukan yaitu:

- a. Melakukan analisis kemungkinan migrasi antara dua *cloud provider* dengan mencari dan mempelajari *tools* yang tersedia untuk melakukan transformasi atau konversi *image* sistem yang didukung oleh dua *cloud provider* baik secara otomatis maupun manual.
- b. Melakukan analisis perbandingan teknologi yang digunakan diantara dua cloud provider seperti jenis teknologi *hypervisor* yang digunakan, sistem operasi yang didukung oleh masing-masing *cloud provider*.
- c. Merencanakan skenario dan metode migrasi yang akan dilakukan
- d. Merencanakan skenario pengujian migrasi
- e. Melakukan analisis ukuran untuk menentukan tingkat interoperabilitas aplikasi diantara dua *cloud provider*

2. Verifikasi Fungsionalitas Aplikasi Sebelum Migrasi

Dalam tahapan ini pengujian fungsionalitas aplikasi akan dilakukan sebagai parameter acuan atau pembandingan aplikasi setelah dilakukan proses migrasi.

3. Migrasi Sistem

Dalam tahapan ini akan dilakukan proses migrasi sistem sesuai dengan skenario dan metode yang telah dirancang di tahap perencanaan.

4. Pengujian dan Verifikasi Fungsionalitas Sistem

Setelah proses migrasi berhasil, selanjutnya akan dilakukan tahap pengujian fungsionalitas seluruh komponen aplikasi sesuai dengan skenario pengujian yang telah dibuat

5. Analisa Hasil Uji

Seluruh hasil pengujian dan verifikasi akan dikumpulkan, dan dianalisis kemudian akan ditentukan tingkat interoperabilitasnya dari tiap *cloud provider* sumber ke *cloud provider* tujuan.

Halaman ini sengaja dikosongkan

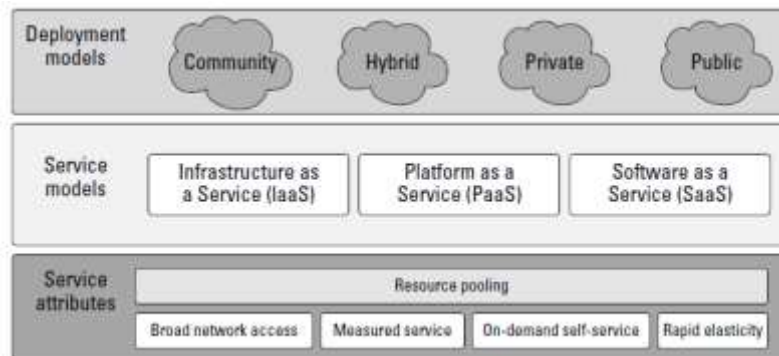
BAB 2

KAJIAN PUSTAKA

2.1 Cloud Computing

Cloud computing merupakan tren baru dalam bidang teknologi informasi yang memindahkan komputasi dan penyimpanan data dari perangkat komputer fisik ke *data center* yang besar (Lenk dkk, 2014). Barry Sosinsky (Sosinky, 2011) dalam bukunya mendefinisikan *cloud computing* sebagai kumpulan aplikasi dan layanan yang berjalan dan beroperasi dalam jaringan terdistribusi atau tersebar dengan memanfaatkan sumber daya *virtual* dan standar *internet*. Sejarah *cloud computing* berawal dari layanan baru perusahaan telekomunikasi yang menawarkan produk VPN (Virtual Private Network) yang sebelumnya merupakan sirkuit data yang bersifat point-to-point (Jadeja, 2012). Kekurangan layanan point-to-point ini adalah mahal dan boros *bandwidth*, sehingga muncul teknologi VPN sebagai layanan alternatif dimana kualitas yang ditawarkan sama dengan *point-to-point* tetapi jauh lebih murah jika ditinjau dari segi biaya.

Terdapat 2 tipe model cloud computing (Sosinky, 2011), yaitu *deployment model* dan *service model*. *Deployment model* merupakan model *cloud computing* yang menjelaskan lokasi, tujuan, dan pengelolaan infrastruktur *cloud*. Sedangkan *service model* terdiri dari berbagai macam tipe layanan *cloud computing* yang ditawarkan. Model ini diadopsi dari NIST(National Institute of Standards and Technology) yang memisahkan *cloud computing* menjadi 2 layanan seperti yang telah disebutkan.



Gambar 2.1 Model Cloud Computing NIST (Sosinky, 2011)

NIST membagi *deployment model* menjadi 4 tipe yaitu :

1. Public Cloud : Jenis infrastruktur *cloud* yang bertujuan untuk penggunaan publik
2. Private Cloud : Jenis *cloud* yang beroperasi khusus untuk sebuah organisasi atau perusahaan
3. Hybrid Cloud : Merupakan infrastruktur campuran atau gabungan dari *private* dan *public cloud*
4. Community Cloud : Community Cloud merupakan *cloud* yang dibangun khusus dengan tujuan tertentu. Contohnya yaitu satu atau beberapa organisasi yang berbagi kepentingan seperti kebijakan, keamanan, regulasi dll.

Sedangkan untuk *service models* dibagi menjadi 3 tipe (Sosinky, 2011) yaitu :

1. IaaS (Infrastructure as a Service) : menyediakan infrastruktur komputasi yang berbasis internet meliputi mesin virtual, *storage* virtual, dan seluruh *hardware* yang dapat diakses oleh pengguna. Penyedia mengelola keseluruhan infrastruktur, sedangkan user mengelola dari segi pembangunan.

Contoh penyedia layanan IaaS :

- Amazon Elastic Compute Cloud (EC2)
- Eucalyptus
- GoGrid
- FlexiScale

- Linode
 - RackSpace Cloud
 - Terremark
2. PaaS (Platform as a Service): memungkinkan pengguna dapat membangun dan menggunakan aplikasi sendiri dalam sistem cloud yang dibangun menggunakan bahasa pemrograman dan perangkat yang didukung oleh penyedia layanan. Dari segi pengelolaan, pengelola PaaS bertanggung jawab dalam infrastruktur cloud, sistem operasi, sedangkan pengguna melakukan penginstalan dan pengelolaan aplikasi.

Contoh penyedia layanan PaaS :

- Force.com
 - GoGrid CloudCenter
 - Google AppEngine
 - Windows Azure Platform
3. SaaS (Software as a Service): mendukung dan menyediakan layanan lengkap mulai dari aplikasi, pengelolaan, dan *user interface* menggunakan *browser*. Pengguna dapat menggunakan layanan *software* yang berjalan dalam lingkungan *cloud* sesuai kebutuhan dan tidak bertanggungjawab dalam sisi penginstalan dan perawatan *software*.

Contoh penyedia layanan SaaS :

- GoogleApps
- Oracle On Demand
- Salesforce.com
- SQL Azure

Terdapat beberapa karakteristik dari *cloud computing* yaitu (Sosinky, 2011) :

1. On-demand Self-service : pengguna yang menyewa layanan dalam *cloud computing* dapat mengakses sumber daya komputasi yang disewa tanpa perlu interaksi dengan pengelola penyedia layanan.

2. Broad Network Access : salah satu karakteristik *cloud computing* yaitu ketersediaan akses jaringan yang mendukung berbagai tipe pengaksesan dan sistem operasi yang digunakan oleh pengguna seperti laptop, handphone, dan PDA.
3. Resource Pooling : Dalam pengelolaan layanan, penyedia menyatukan seluruh sumber daya yang dapat dipakai oleh banyak pengguna atau penyewa. Konsep abstraksi digunakan sehingga user tidak mengetahui posisi fisik sumber daya yang sedang diakses, juga tidak dapat membedakan apakah sumber daya yang dipakai memakai perangkat fisik atau virtual
4. Rapid Elasticity : Sumber daya sistem dapat diperluas secara elastis sesuai kebutuhan pengguna baik dari segi perangkat fisik seperti penambahan komputer yang lebih powerful.
5. Measured Service : Penggunaan sumber daya dalam cloud dapat diukur, diaudit, dan dilaporkan ke pengguna berdasarkan meter system. Pengguna membayar sesuai dengan layanan yang disewa.

Terdapat beberapa manfaat yang dapat diperoleh dalam penggunaan *cloud computing* (Sosinky, 2011):

1. Penggunaan *cloud computing* dapat menekan biaya operasional karena sistem beroperasi dengan efisiensi dan pemanfaatan sumber daya yang lebih besar.
2. Tidak memerlukan lisensi *software* dan *hardware* untuk membangun sistem yang diinginkan.
3. Adanya jaminan QOS(Quality of Service) dari penyedia layanan *cloud computing*.
4. Beberapa fitur seperti *load balancing* dan *failover* menjamin kehandalan dalam layanan *cloud computing*
5. Dengan adanya pihak lain yang mengelola infrastruktur sistem dalam *cloud*, perusahaan yang menyewa layanan *cloud computing* dapat fokus ke pengelolaan bisnis yang berjalan.

6. Pemeliharaan sistem dan infrastruktur menjadi lebih sederhana dan mudah karena sistem tersentralisasi

Selain berbagai macam keuntungan diatas, penggunaan layanan *cloud computing* juga memiliki beberapa kekurangan (Sosinky, 2011) yaitu :

1. Meskipun layanan *cloud computing* mendukung elastisitas sesuai proses bisnis dan keinginan pengguna, tetapi tidak semua keinginan pengguna akan dilayani dan diimplementasikan.
2. Seluruh aplikasi *cloud computing* memiliki masalah yang sama dalam hal koneksi jaringan internet khususnya latensi. Model *cloud computing* tidak cocok ketika aplikasi yang digunakan oleh pengguna membutuhkan transfer data dalam jumlah besar.

2.1.1 Amazon EC2

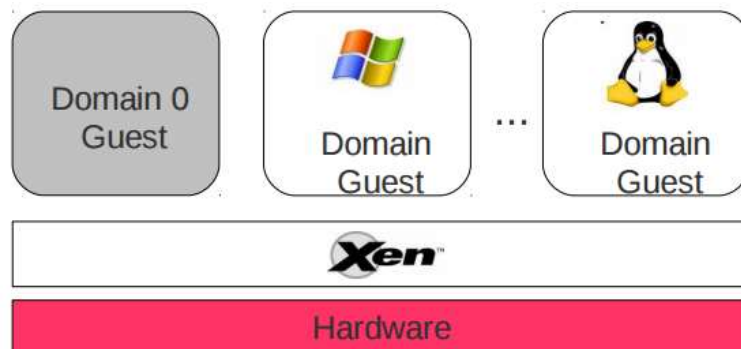
Amazon EC2 atau Amazon Elastic Compute Cloud merupakan salah satu penyedia layanan infrastruktur *cloud computing* komersial yang terintegrasi yang diperuntukkan untuk individu atau perusahaan. Amazon EC2 berbentuk platform komputasi berupa virtual komputer yang dapat dikostumisasi maupun dikembangkan dengan menggunakan prinsip *cluster* dan *load balance* (Amazon Web Services, 2016).

Beberapa fitur yang disediakan oleh Amazon EC2 yaitu :

1. Lingkungan *virtual computing* atau komputasi virtual
2. Menyediakan *preconfigured templates* yang disebut Amazon Machine Images (AMIs) dimana dapat digunakan untuk membangun mesin virtual, dapat berupa sistem operasi atau software tambahan lainnya.
3. Berbagai macam jenis konfigurasi CPU, memory, storage, networking untuk kostumisasi mesin virtual
4. Layanan *secure login* menggunakan *key pairs*
5. *Storage volume* untuk data sementara dimana akan dihapus jika mesin virtual yang telah dibangun dihentikan/dihapus

6. *Persistent storage volumes* menggunakan Amazon Elastic Block Store (Amazon EBS)
7. Tersedianya berbagai lokasi fisik untuk resource yang kita gunakan
8. Firewall memungkinkan untuk mengatur protocol, port, dan range IP yang mengakses sumber daya cloud kita
9. Alamat IPv4 statis untuk cloud computing dinamis
10. Metadata
11. Virtual network yang memungkinkan untuk melakukan isolasi secara logic dari resource Amazon EC2 lain

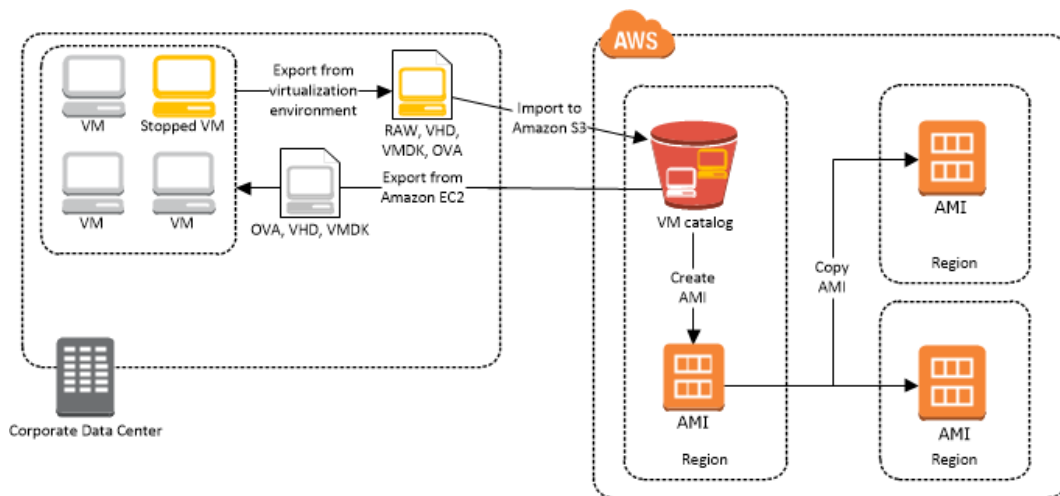
Amazon EC2 menggunakan hypervisor tipe XEN yang dikustom untuk pembangunan seluruh mesin virtual yang dimilikinya.



Gambar 2.2 Amazon EC2 Hypervisor (Badola, 2015)

Amazon EC2 menggunakan *hypervisor* tipe Xen Bare Metal dalam infrastrukturnya, dimana Xen mendukung 2 tipe virtualisasi yaitu HVM atau bisa disebut Hardware Virtual Machine dan PV atau Paravirtualization (Badola,2015).

Dalam mendukung interoperabilitas, Amazon EC2 mendukung migrasi ke dalam sistemnya melalui beberapa cara salah satunya yaitu VM Import/Export dimana pengguna dapat melakukan migrasi sistem yang dimiliki ke sistem Amazon EC2.



Gambar 2.3 Fitur Amazon VM Export/Import (Amazon Web Services, 2016)

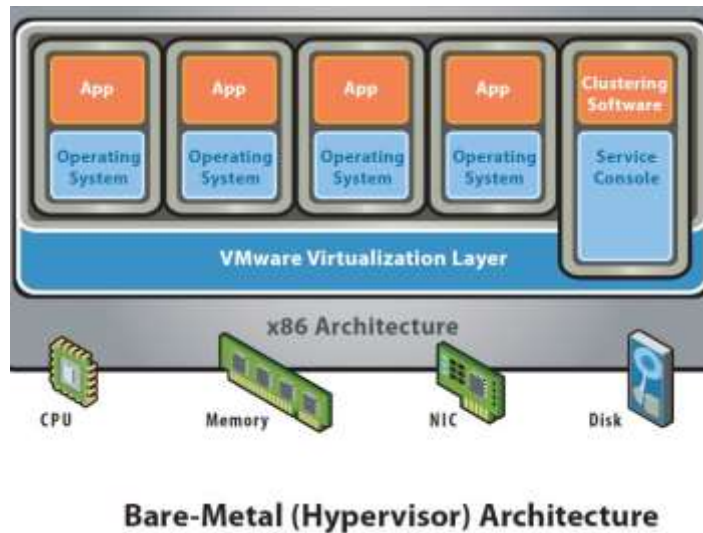
Alur proses mekanisme untuk melakukan Import VM pada Amazon EC2 yaitu melakukan bundle VM *image* menjadi beberapa bentuk format yang didukung oleh Amazon EC2 yaitu RAW, VHD, VMDK, dan OVA kemudian file tersebut diupload ke Amazon S3. Ketika proses import berhasil maka image akan terdaftar dalam katalog kemudian ditransformasi menjadi AMI (Amazon Machine Image) yang selanjutnya dibedakan menurut *region* yang ditentukan. Sedangkan untuk proses Export VM catalog yang telah ada akan dibundle menjadi beberapa format file yaitu OVA, VHD, dan VMDK.

Untuk dapat mengakses sumber daya sistem yang ada di Amazon EC2 seperti login, Amazon EC2 menggunakan sistem keamanan kriptografi untuk melakukan enkrip dan dekrip informasi login. Kriptografi public key digunakan untuk melakukan enkrip data seperti password kemudian penerima menggunakan *private key* untuk melakukan dekrip data. Kedua kombinasi *public key* dan *private key* tersebut disebut *key pairs*.

2.1.2 Indonesian Cloud

Indonesian Cloud merupakan salah satu penyedia layanan *cloud* di Indonesia berlisensi VMWare VCloud Powered dimana didukung oleh teknologi komputasi awan dan virtualisasi terkemuka dari VMware yaitu, VMware® vSphere dan VMware vCloud Director yang mempunyai layanan cloud server dimana menawarkan hosting infrastruktur cloud dengan jaminan *no single points*

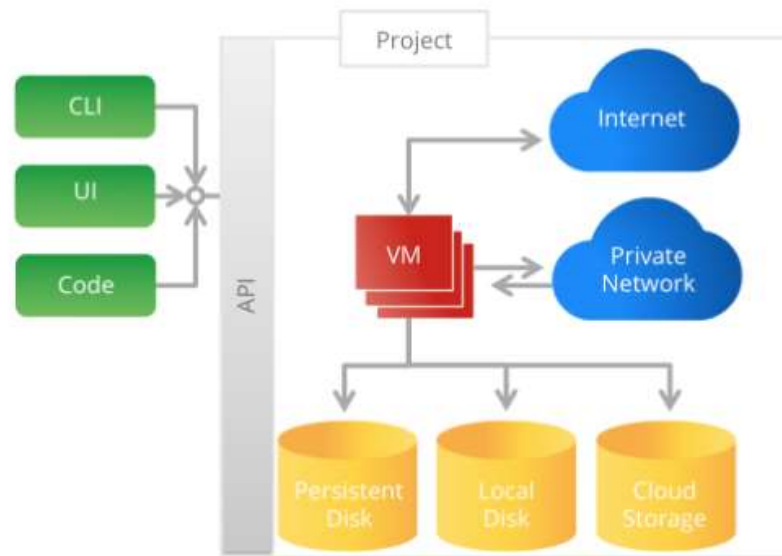
of failure (Indonesian Cloud, 2016). vCloud Air merupakan *secure cloud* IaaS yang dibangun menggunakan fondasi *hypervisor* VMware vSphere.



Gambar 2.4 Arsitektur Hypervisor VMWare (Amazon Web Services, 2016)

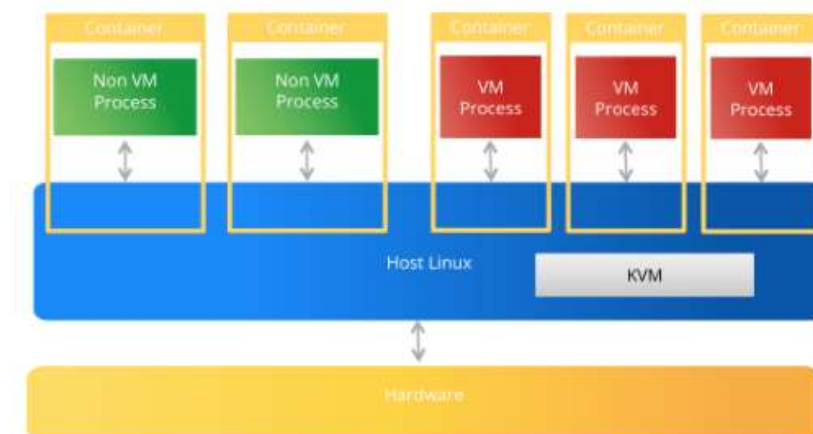
2.1.3 Google Compute Engine

Merupakan komponen bertipe IaaS dari Google Cloud Platform yang memungkinkan untuk menjalankan mesin virtual atau Virtual Machine (VM) dari *image* yang disediakan secara default maupun custom dari user diatas mesin fisik Google.



Gambar 2.5 Arsitektur layanan Google Compute Engine

Gambar 2.5 menunjukkan arsitektur layanan Virtual Machine pada Google Cloud Engine. Sedangkan teknologi *hypervisor* KVM (Kernel Based Virtual Machine).

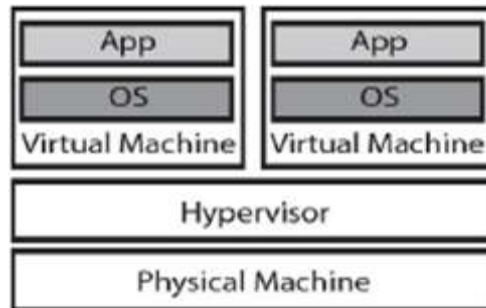


Gambar 2.6 Arsitektur Hypervisor Google Cloud Engine

2.2 Teknologi Virtualisasi

Virtualisasi merupakan teknologi utama yang memungkinkan munculnya paradigma *cloud computing*. Teknik virtualisasi memungkinkan beberapa sistem operasi berjalan secara simultan di dalam satu mesin fisik (Strunk dan Dargie,

2013). Virtualisasi memisahkan aplikasi, desktop, mesin, jaringan, data, dan layanan dari sumber daya fisik (PCI Security Standard Council, 2011).



Gambar 2.7 Prinsip Virtualisasi

Cloud computing berasal dari 2 konsep penting yaitu konsep abstraksi dimana menyembunyikan detail implementasi sistem seperti pengguna tidak mengetahui dimana letak perangkat fisik yang diakses dan konsep virtualisasi dimana menyatukan dan membagi sumber daya yang tersedia (Sosinky, 2011)..

Beberapa tipe virtualisasi dalam lingkungan *cloud computing* yaitu :

1. Akses : Pengguna atau penyewa layanan *cloud computing* dapat melakukan permintaan akses dari suatu layanan *cloud* dari lokasi yang berbeda
2. Aplikasi : Cloud mengelola banyak aplikasi dan pengguna dapat melakukan permintaan untuk terhubung langsung sesuai kondisi.
3. CPU : Data komputer fisik dapat dibagi menjadi beberapa virtual mesin yang berjalan bersamaan dimana masing-masing mesin mempunyai beban kerja yang berbeda.
4. Storage : Data tersimpan di perangkat penyimpanan dan direplikasi untuk menyediakan layanan redundansi.

Selain tipe virtualisasi dalam cloud computing, terdapat beberapa tipe virtualisasi lain (PCI Security Standard Council, 2011) yaitu :

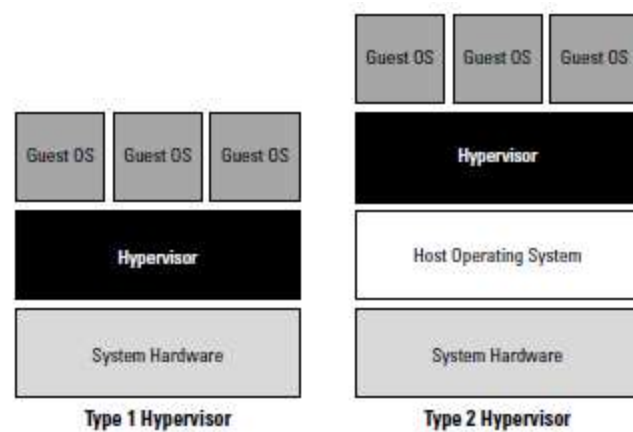
1. Virtualisasi Sistem Operasi

Umumnya digunakan untuk mengelola sumber daya dalam sistem operasi yang terletak di satu server fisik serta membagi sumber daya tersebut menjadi bagian-bagian yang lebih kecil dan masih terletak dalam kernel OS yang sama tetapi berbeda *library* atau distribusi seperti sumber daya virtual, aplikasi virtual, dan server privat virtual.

2. Virtualisasi Hardware/Platform

Salah satu contoh dari virtualisasi hardware yaitu teknologi partisi *hardware* atau bisa disebut *hypervisor*. Terdapat 2 jenis *hypervisor* (Sosinky, 2011) yaitu :

- a. Type 1 VM bisa juga disebut Native VM atau Bare Metal VM dimana sistem operasi yang dijalankan dalam virtual mesin disebut *guest operating sistem* atau sistem operasi tamu serta dapat menjalankan beberapa tamu dalam satu mesin fisik. Sistem operasi yang berjalan dapat melakukan virtualisasi penuh dan *hypervisor* tipe ini tidak mempunyai sistem operasi *host*. Contoh produk *hypervisor* tipe 1 yaitu LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Server, VirtualLogix VLX, VMware ESX and ESXi, dan Wind River VxWorks
- b. Type 2 VM yaitu tipe *hypervisor* yang diinstal diluar sistem operasi. Dalam tipe ini operasi I/O berada di luar lingkungan virtual sehingga lebih mudah dari segi pemrograman serta lebih efisien untuk menjalankan perangkat I/O karena berada di luar lingkungan virtual. Contoh produk dari tipe ini yaitu Microsoft's Hyper-V dan Xen.



Gambar 2.8 Perbedaan 2 Jenis Hypervisor (Sosinky, 2011)

c. Virtualisasi jaringan

Virtualisasi jaringan membedakan komponen *logic* dalam jaringan fisik. Seluruh komponen fisik jaringan seperti *router*, *switch*, *firewall* terdapat komponen *logic* yang berfungsi sebagai perangkat virtual.

Beberapa tipe komponen logic dalam perangkat jaringan yaitu :

1. Data plane : berfungsi melakukan *forward* komunikasi data diantara host dalam suatu jaringan
2. Control Plane : berfungsi mengelola *traffic*, jaringan, dan informasi *routing*
3. Management Plane : mengelola komunikasi langsung diantara 2 perangkat dengan tujuan mengatur dan mengelola perangkat manajemen perangkat jaringan

d. Data Storage

Virtualisasi penyimpanan data mengelola beberapa perangkat penyimpanan fisik yang bergabung menjadi satu perangkat menjadi sebuah subuah perangkat penyimpanan

e. Memory

Merupakan penggabungan memory fisik yang tersedia dari beberapa sistem untuk membuat sebuah kolam memory virtual yang dapat diakses antar komponen sistem

Salah satu komponen dari sistem virtualisasi yaitu *virtual machine* (PCI Security Standard Council, 2011). *Virtual machine*(VM) yaitu *software* komputer yang beroperasi layaknya komputer fisik yang menjalankan sistem operasi dan aplikasi (VMWare, 2015). *Virtual machine* berjalan secara paralel dan berbagi sumber daya dengan VM lainnya. Komponen yang mengatur dan menjalankan beberapa VM disebut *hypervisor* atau virtual machine monitor(VMM).

Dalam lingkungan virtualisasi terdapat suatu mekanisme yang menyediakan aspek portabilitas dalam melakukan pembangunan sistem dalam cloud menggunakan *image system* (Sosinky, 2011). *Image system* mempunyai kemampuan untuk melakukan pengkopian dan pengandaan seluruh komponen sistem dalam satu file. File image ini berisi sistem operasi, driver perangkat, aplikasi, informasi status sistem, file data yang dapat digunakan untuk melakukan *restore* sistem.

Virtual machine image menyediakan fasilitas pembangunan dan perbaikan secara cepat sistem virtual di beberapa *host* yang berbeda (PCI Security Standard Council, 2011). VM *image* menyimpan data-data sensitif sistem ketika *image* tersebut diambil seperti data konten aktif *memory* yang dapat mengakibatkan proses pengambilan komponen sistem yang kurang lengkap seperti penyimpanan data atau informasi penting lainnya.

Komponen file virtual machine :

1. .XML : File xml berisi detail konfigurasi di setiap *virtual machine*
2. .BIN : File bin berisi *memory virtual machine* atau *snapshot* dari status terakhir yang tersimpan
3. .VSV : File vsv berisi status tersimpan dari peralatan-peralatan yang berhubungan dengan *virtual machine*
4. .VHD : Berisi file *virtual hardisk* dari *virtual machine*

5. AVHD : file disk berbeda yang digunakan dalam pembuatan *virtual machine snapshot*

2.3 Migrasi Cloud

Migrasi *virtual machine* merupakan salah satu kemampuan virtualisasi sistem dimana memungkinkan suatu aplikasi dapat dipindahkan secara transparan dari satu host fisik ke host fisik lainnya tanpa kehilangan fitur yang dimiliki (Strunk dan Dargie, 2013). Secara umum migrasi dilakukan dengan memindah/mentransfer aplikasi beserta seluruh sistem *virtual machine* termasuk CPU, *memory*, *disk* dari sistem asal ke sistem tujuan. Selain dimanfaatkan untuk mengelola aplikasi beserta sumber daya dalam lingkungan *data center* dan sistem *cloud* yang tervirtualisasi, migrasi *virtual machine* juga memungkinkan untuk merelokasi sistem secara dinamis ke sistem lain yang memiliki eksekusi yang lebih cepat dan lebih handal (Open Data Center Alliance, 2013).

DMTF mendefinisikan tiga tipe operasi migrasi untuk *virtual machine* (Open Data Center Alliance, 2013):

1. Level 1 : VM hanya berjalan di produk virtualisasi/arsitektur CPU/virtual hardware tertentu. Migrasi level 1 setara dengan operasi ‘suspend’ di sistem awal/sumber dan operasi ‘resume’ di sistem tujuan/target. Mekanisme ‘Live Migration’ dimungkinkan berjalan di Level 1.
2. Level 2 : VM berjalan di keluarga perangkat *virtual* yang lebih spesifik. Migrasi level 2 setara dengan operasi ‘shut-down’ di sistem sumber diikuti dengan ‘reboot’ di sistem target. Migrasi berbeda *hypervisor* dimungkinkan berjalan pada level ini.
3. Level 3 : VM dapat berjalan di banyak keluarga perangkat virtual. Level ini merupakan framework paling fleksibel untuk migrasi VM, tetapi membutuhkan pembangunan yang terintegrasi dan teknologi yang digunakan belum familiar.

Beberapa metodologi dalam migrasi *cloud* (Pappas, 2014) yaitu:

1. Cloud-RMM (Cloud-Reference Migration Model)

Merupakan model konseptual yang menggunakan pendekatan bottom-up untuk mengidentifikasi aktivitas low-level dan teknik yang kemudian dikategorisasikan untuk membentuk proses. Cloud-RMM juga menggunakan pendekatan top down untuk mementik framework yang terdiri dari beberapa proses, aktifitas, dan teknik. Tabel 1 mendeskripsikan tahapan metodologi migrasi Cloud-RMM.

Tabel 2.1 Metodologi Cloud-RMM (Pappas, 2014)

Proscess	Task
Planning	Feasibility study, migration requirement analysis, decisions of provider and services, migration strategies
Execution	Code modification, architecture extraction, data extraction, transformation
Evaluation	Deployment, Testing, Validation
Crosscutting concerns	Governance, Security, Training, Effort estimation,organizational change, Multitenancy

2. ARTIST (Advanced software-based service provisioning and migration of legacy software)

Merupakan pendekatan yang berfokus pada aplikasi non cloud. ARTIST membangun perangkat dan metode untuk mengatasi masalah yang signifikan serta mengurangi biaya dan resiko. Tabel 2 menjelaskan tahapan proses dalam metodologi ARTIST.

Tabel 2.2 Metodologi ARTIST(Pappas, 2014)

Proscess	Task
Pre-migration	Migration Feasibility Assesment
Migration	Application Discovery & Understanding and Migration
Provisioning	Testing, Verification, & Certification

2.4 Interoperabilitas

Dalam proses migrasi dari satu sistem ke sistem lainnya terutama dalam lingkungan *cloud computing* kemampuan interoperabilitas menjadi faktor yang penting. Interoperabilitas *cloud* mengacu pada kemudahan migrasi dan integrasi aplikasi,data, dan beban kerja diantara penyedia layanan *cloud* (Dowell dkk, 2011). Selain itu adanya incompatibilitas standar diantara penyedia *cloud platform provider* mengakibatkan migrasi aplikasi dari suatu *provider* ke *provider* lain masih menjadi proses yang sulit (Gholami dkk, 2016)

LISI (Levels of Information System Interoperability) (Dowell dkk, 2011) mengklasifikasikan atribut pertukaran dan pembagian informasi dan layanan antar sistem yaitu :

1. Prosedur

Mencerminkan tingkat interoperabilitas yang dihasilkan dari kebijakan operasional, panduan pembangunan fungsi program, standar arsitektur sistem seperti standar *hardware*, *software* sistem, komunikasi, dan aplikasi.

2. Aplikasi

Mencerminkan kemampuan *software* aplikasi yang dapat bekerja di sistem dan *platform* yang berbeda.

3. Infrastruktur

Mencerminkan tingkat konektifitas antara sistem dan aplikasi serta cara sistem berinteraksi satu sama lain.

4. Data

Mencerminkan kemampuan fleksibilitas format data dan kekayaan informasi yang dipertukarkan di seluruh sistem dan *domain*.

Vendor lock-in merupakan merupakan keadaan dimana pengguna suatu produk bergantung pada satu *provider* dan tidak dapat berpindah ke *provider* lain tanpa biaya yang besar dan waktu yang lama (Opera-Martins dkk, 2014). Salah satu cara untuk mengatasi masalah *vendor lock-in* yaitu pembuatan standar interoperabilitas untuk mendukung mudahnya migrasi beban kerja, aplikasi, data dari satu cloud provider ke provider lainnya (Lewis, 2013).

Menurut (Cloud Standards Customer Council, 2013) salah satu isu utama dalam *cloud computing* yaitu *cloud service provider lock-in*, dimana ketika suatu layanan *cloud* dari suatu *cloud provider* telah diadopsi, akan sangat sulit jika akan berganti menggunakan layanan *cloud* yang setara di *cloud provider* lain. Munculnya berbagai macam standar akan meningkatkan portabilitas dan interoperabilitas sistem diantara berbagai macam *cloud provider* dan nantinya akan mengurangi permasalahan *cloud service provider lock-in* ini.

Paper (Lewis, 2013) mendefinisikan 4 dasar interoperabilitas kasus dalam lingkungan *cloud computing*:

1. Otentikasi user

Setiap pengguna harus teridentifikasi dalam lingkungan cloud. Beberapa standar yang mendukung kasus ini yaitu :

- a. Amazon Web Services Identity Access Management (AWS IAM) merupakan mekanisme otentifikasi dan manajemen user yang digunakan oleh Amazon yang mendukung pembuatan user beserta manajemen perizinan beberapa user dalam satu akun AWS. Selain Amazon, Eucalyptus juga menggunakan standar ini untuk otentikasi user.
- b. OAuth merupakan open protocol yang dibuat oleh IETF(Internet Engineering Task Force) yang menyediakan metode untuk pengguna untuk mengakses sumber daya server atas nama pemilik sumber daya.

- c. OpenID merupakan open standard yang memungkinkan pengguna terotentikasi dengan metode terdesentralisasi
 - d. WS-Security merupakan standar keamanan OASIS yang mengamankan pesan bertipe SOAP menggunakan XML Signature dan XML Encryption
2. Migrasi Beban Kerja
- Migrasi beban kerja merepresentasikan migrasi VM *image* dari satu *cloud provider* ke *cloud provider* lain. Tipe migrasi beban kerja membutuhkan ekstraksi beban kerja di cloud provider sumber dan upload beban kerja tersebut di *cloud provider* lain. Beberapa standar yang mendukung yaitu :
- a. Amazon Machine Image(AMI) merupakan jenis VM tertentu yang dapat dibangun dalam lingkungan Amazon EC2. Eucalyptus dan OpenStack juga mendukung tipe AMI.
 - b. Open Virtualization Framework(OVF) merupakan standar pembangunan VM yang didukung oleh DMTF. Beberapa provider cloud yang mendukung yaitu Amazon EC2, Eucalyptus, dan OpenStack
 - c. Virtual Hard Disk(VHD) merupakan format VM yang didukung oleh Microsoft. Platform *cloud* yang mendukung VHD yaitu Amazon EC2 dan Microsoft Azure.
3. Migrasi Data
- Dalam konteks interoperabilitas, ketika data telah dipindah ke cloud provider yang baru, seluruh program yang melakukan operasi CRUD(create, retrieve, update, delete) di *cloud* sebelumnya harus dapat melakukan operasi tersebut di cloud provider yang baru. Beberapa standar yang mendukung migrasi data yaitu :
- a. Cloud Data Management Interface(CDMI) merupakan standar yang didukung oleh SNIA(Storage Networking Industry Association) dimana mendefinisikan API untuk operasi CRUD pada elemen data di lingkungan cloud.
 - b. SOAP
 - c. REST

4. Menejemen workload

Merupakan menejemen beban kerja sistem yang dibangun dalam lingkungan cloud.

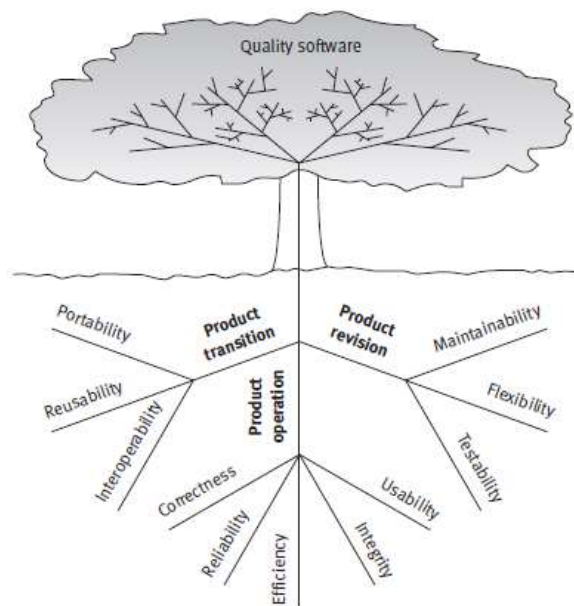
Terdapat beberapa faktor yang mempengaruhi aspek bisnis dari Interoperabilitas Virtual Machine yaitu (Open Data Center Alliance, 2013) :

1. Migrasi Aplikasi : merupakan kemampuan memindahkan aplikasi dari satu mesin Virtual Mesin di satu lingkungan *cloud provider* ke *cloud provider* lain dengan usaha minimal dengan pertimbangan efisiensi biaya, fungsionalitas, level layanan dll
2. Pengembangan Private Cloud : penambahan sumber daya komputasi pada jaringan *cloud private* lokal dengan mudah seperti kemudahan pengelolaan dan migrasi beban kerja antara lingkungan *cloud privat* dan *cloud public*
3. Keberlangsungan Bisnis : proses migrasi atau replikasi aplikasi diantara *provider* menyebabkan beberapa masalah dan gangguan salah satunya yaitu pelanggaran keamanan.

2.5 Pengujian Aplikasi

Dalam dunia pengujian aplikasi, faktor kualitas software merupakan komponen yang tidak dapat dipisahkan. Terdapat beberapa model faktor penentu kualitas software, salah satunya yaitu model McCall (Galin, 2004) yang terdiri dari 11 faktor diantaranya yang dikategorikan dalam 3 kelompok yaitu:

1. Product operation factors: Correctness, Reliability, Efficiency, Integrity, Usability.
2. Product revision factors: Maintainability, Flexibility, Testability.
3. Product transition factors: Portability, Reusability, Interoperability



Gambar 2.9 Model McCall (Galin, 2004)

Fokus utama pengujian aplikasi cloud dalam penelitian ini yaitu faktor interoperability yang dalam konteks (Galin, 2004) disebut *portability* dimana merupakan kemampuan adaptasi dari sistem *software* ketika dipindah ke lingkungan lain yang berbeda *hardware*, sistem operasi, dan sebagainya.

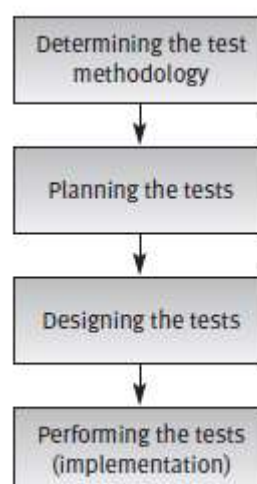
Pengujian aplikasi atau bisa disebut *software test* merupakan komponen dari *software quality assurance* yang bertujuan untuk melakukan review terhadap *software* yang telah berjalan. Pengujian aplikasi yang dilakukan berdasarkan pada list *test case* yang telah disiapkan yang merepresentasikan berbagai macam skenario yang diinginkan. Dalam pengujian aplikasi juga dikenal istilah *white box* dan *black box testing*.

IEEE (1990) mendefinisikan *black box testing* yaitu pengujian dimana menghiraukan mekanisme internal dari sistem atau komponen dan hanya berfokus pada output yang terjadi sebagai respon input yang dipilih dan kondisi yang dieksekusi. Definisi lain dari *black box testing* yaitu pengujian yang diselenggarakan untuk mengevaluasi kinerja sistem atau komponen dengan kebutuhan fungsional tertentu. Sedangkan *white box testing* yaitu pengujian yang dilakukan sampai ke mekanisme internal dari sistem atau komponen. Realisasi

dari *white box testing* ini membutuhkan verifikasi dari setiap baris program. Terlihat pada tabel di bawah bahwa untuk *portability test* cukup menggunakan *black box testing*.

Test classification according to requirements	White box testing	Black box testing
1.1 Output correctness tests		+
1.2 Documentation tests		+
1.3 Availability (reaction time) tests		+
1.4 Data processing and calculations correctness tests	+	
1.5 Software qualification tests	+	
2. Reliability tests		+
3. Stress tests (load tests and durability tests)		+
4. Software system security tests		+
5.1 Training usability tests		+
5.2 Operational usability tests		+
6. Maintainability tests	+	+
7. Flexibility tests		+
8. Testability tests		+
9. Portability tests		+
10. Reusability tests	+	
11.1 Software interoperability tests		+
11.2 Equipment interoperability tests		+

Dalam melakukan proses pengujian juga terdapat langkah-langkah proses yang diikuti.



Gambar 2.10 Proses Pengujian (Galin. D, 2004)

Tahap pertama yaitu menentukan metodologi pengujian termasuk strategi pengujian yang akan digunakan apakah menggunakan big bag atau incremental testing. Selanjutnya yaitu melakukan merencanakan pengujian apakah unit test, integration test, maupun system test. Unit test mencakup pengujian unit kecil dari software atau modul, *integration* test mencakup gabungan dari beberapa unit yang bergabung menjadi sub sistem, sedangkan sistem test mengacu pada keseluruhan sistem.

Pada banyak perusahaan proses pengujian terdiri unit testing kemudian *black box testing* yang terdiri dari *build acceptance test*, fungsional test, system test, dan *reliability test*. Salah satu teknik *black box testing* yang digunakan untuk menguji dan memvalidasi software berbasis cloud yaitu BAT (Build Acceptance Test). Pengujian aplikasi menggunakan BAT menggunakan 2 metrik yaitu melakukan pemeriksaan jumlah layanan yang terlibat dalam *use case* yang dapat mempengaruhi *test case* dalam menemukan kesalahan dan mencari layanan yang spesifik berhubungan dengan banyaknya kesalahan pada *software* sebelumnya.

Selain itu (Srikanth dkk, 2016) dalam jurnalnya menjelaskan tentang teknik membagi prioritas test case, salah satunya yaitu *total greedy prioritization technique* dimana memilih test case di setiap step yang mewakili keseluruhan elemen.

Secara umum terdapat beberapa pengujian untuk aplikasi *website* (Cai dkk, 2014), diantaranya yaitu:

1. Unit Testing: Pengujian yang berfokus bagian kecil dari kode seperti *class*, *method*, *function* atau *interface* untuk memastikan kode yang telah dibuat berfungsi.
2. Functional Testing: Proses pengujian yang memastikan aplikasi website berjalan seperti yang diinginkan dan melakukan verifikasi apakah aplikasi tersebut dapat menampilkan apa yang diinginkan pengguna. Pengujian fungsionalitas juga memeriksa kelayakan aplikasi apakah sesuai dengan ekspektasi pengguna atau tidak.

3. Compatibility Testing: Pengujian untuk mencari seberapa bagus performansi sistem website ketika dipindah ke lingkungan sistem lain yang berbeda dari segi *hardware*, *network*, sistem operasi, *web browser* dll.
4. Performance Testing: Pengujian untuk mengukur stabilitas, *response time*, *scalability*, dan *throughpuy* dari aplikasi website. Pengujian ini berfokus pada mencari masalah performansi dalam *design* dan arsitektur dari *software*.
5. Load Testing : Pengujian yang bertujuan untuk mengukur reaksi aplikasi ketika berada pada kondisi beban normal dan dalam kondisi beban yang tinggi.
6. Stress Testing : Pengujian untuk mengevaluasi reaksi aplikasi jika berada dalam kondisi diluar spesifikasinya
7. Regression Testing : Bertujuan untuk mencari kembali bugs setelah perubahan komponen sistem
8. Security Testing : Bertujuan untuk memastikan apakah aplikasi yang digunakan dapat mencegah pencurian data atau ancaman lainnya.

Dalam melakukan pengujian fungsionalitas program tentunya akan ditemukan suatu *error/defect software* jika terdapat masalah pada aplikasi. Penggolongan/klasifikasi defect severity terdiri dari 4 klasifikasi.

1. Critical : jika error yang terjadi mempengaruhi fungsionalitas aplikasi yang bersifat kritis. Seperti kegagalan instalasi.
2. Major : jika error yang terjadi mempengaruhi fungsionalitas utama aplikasi yang bersifat major seperti fitur dari suatu modul tidak berjalan sesuai proses bisnisnya.
3. Minor : jika error yang terjadi mempengaruhi fungsionalitas utama aplikasi.
4. Trivial : jika error yang terjadi mempengaruhi fungsionalitas dari aplikasi

2.6 Penelitian Sebelumnya

TIOSA (Testing VM Interoperability at an OS and Application Level) (Lenk dkk, 2014) merupakan suatu metode untuk mengukur interoperabilitas sistem antara dua *hypervisor* yang berbeda. TIOSA menggunakan metode pengetesan berbasis *greybox* yang berfokus pada verifikasi aplikasi dan sistem operasi.

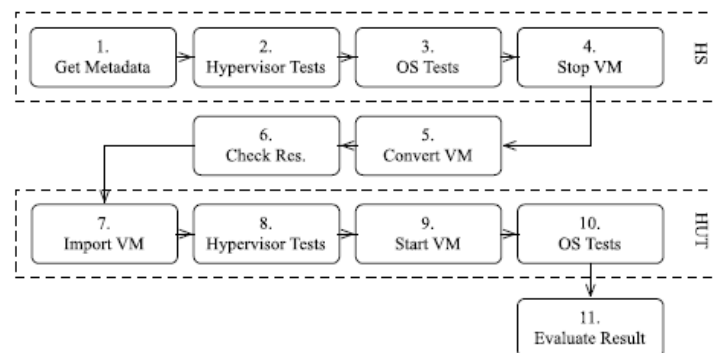
Metode pengujian TIOSA yaitu :

1. Proses model yang terstruktur dan terulang

- a. Melakukan pengecekan interoperabilitas

Menentukan kemungkinan migrasi antara dua *hypervisor* dengan mengumpulkan *metadata* yang dibutuhkan untuk proses migrasi tersedia dan memungkinkan untuk terjadi proses migrasi.

- b. Melakukan proses *copy* VM diantara dua *hypervisor*



Gambar 2.11 Metode TIOSA (Lenk dkk, 2014)

Proses ini memastikan VM berhasil dikonversi dan dapat berjalan di *hypervisor* tujuan. Seluruh hasil test yang telah dilakukan masih dalam batas yang *tolerable*

2. Ukuran untuk menggambarkan hasil tes

Hasil test dikelompokkan dalam kategori Success, Warning, dan Failure

3. Metrik evaluasi pembandingan

Terdapat metrik evaluasi pembandingan untuk menentukan indikator hasil keseluruhan pengujian.

Implementasi metode pengujian TIOSA menggunakan kombinasi dari beberapa produk virtualisasi server yaitu VMWare, Citrix Xen, KVM dan Microsoft Hyper-V. Total kombinasi migrasi berjumlah 12 kombinasi untuk masing-masing sistem operasi. Sistem operasi yang digunakan dalam implementasi yaitu CentOS 6.2, Ubuntu 12.0.4, dan Microsoft Windows 2008 R2 64-bit. Hasil dari pengujian interoperabilitas antar hypervisor pada TIOSA dapat dilihat pada gambar sebagai berikut :

		HYPERVISOR-TARGET			
		VMware	Citrix Xen	KVM	Hyper-V
HYPERVISOR-SOURCE	VMware		Warning	Failure	Failure
	Citrix Xen	Failure		Failure	Failure
	KVM	Failure	Failure		Failure
	Microsoft Hyper-V	Failure	Failure	Failure	

No Result Successful Warning Failure

Gambar 2.12 Hasil Pengujian CentOS 6.6 (Lenk dkk, 2014)

Hasil pengujian pada sistem operasi CentOS menunjukkan bahwa setelah konversi image sistem tidak ada yang menunjukkan hasil Successful, tetapi masih ada yang menunjukkan hasil Warning yaitu konversi Image dari hypervisor VMWare ke Citrix Xen, selain itu sisa pengujianl ainnya masuk ke kategori Failure.

		HYPERVISOR-TARGET			
		VMware	Citrix Xen	KVM	Hyper-V
HYPERVISOR-SOURCE	VMware				
	Citrix Xen				
	KVM				
	Microsoft Hyper-V				

No Result Successful Warning Failure

Gambar 2.13 Hasil Pengujian Ubuntu 12.04 (Lenk dkk, 2014)

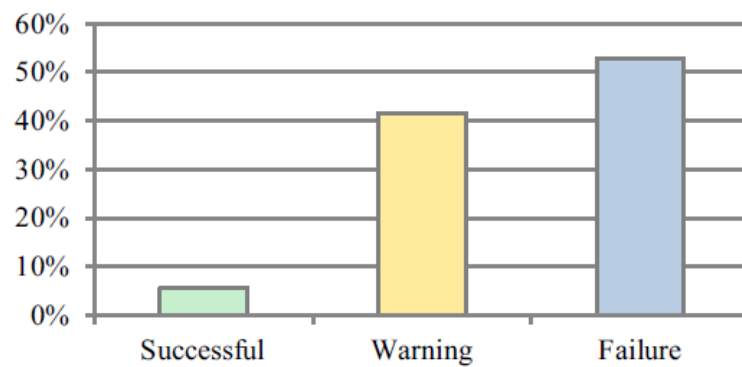
Dari hasil pengujian didapatkan hasil bahwa skenario pengujian konversi image menggunakan sistem operasi Ubuntu 12.04 yang mendapatkan status Successful yaitu dari Citrix Xen ke HyperV.

		HYPERVISOR-TARGET			
		VMware	Citrix Xen	KVM	Hyper-V
HYPERVISOR-SOURCE	VMware				
	Citrix Xen				
	KVM				
	Microsoft Hyper-V				

No Result Successful Warning Failure

Gambar 2.14 Hasil Pengujian Windows Server 2008 R2 (Lenk dkk, 2014)

Dari hasil pengujian didapatkan hasil bahwa skenario pengujian konversi image menggunakan sistem operasi Windows Server 2008 R2 yang mendapatkan status Successful yaitu dari Citrix Xen ke HyperV.



Gambar 2.15 Hasil Keseluruhan (Lenk dkk, 2014)

Dari 36 kemungkinan migrasi image VM yang diuji sebanyak 2 skenario konversi yang berhasil mendapatkan status Successful, 15 mendapatkan status Warning, sedangkan sisanya yaitu 19 konversi image mendapatkan status Failure.

Kekurangan dari metode pengujian TIOSA yaitu hanya menguji interoperabilitas di level *hypervisor* dan sistem operasi, dan belum menguji pada tingkat aplikasi tertentu yang berjalan pada sistem sehingga metode TIOSA belum dapat diimplementasikan secara optimal untuk pengujian interoperabilitas pada sistem yang terdapat di industri.

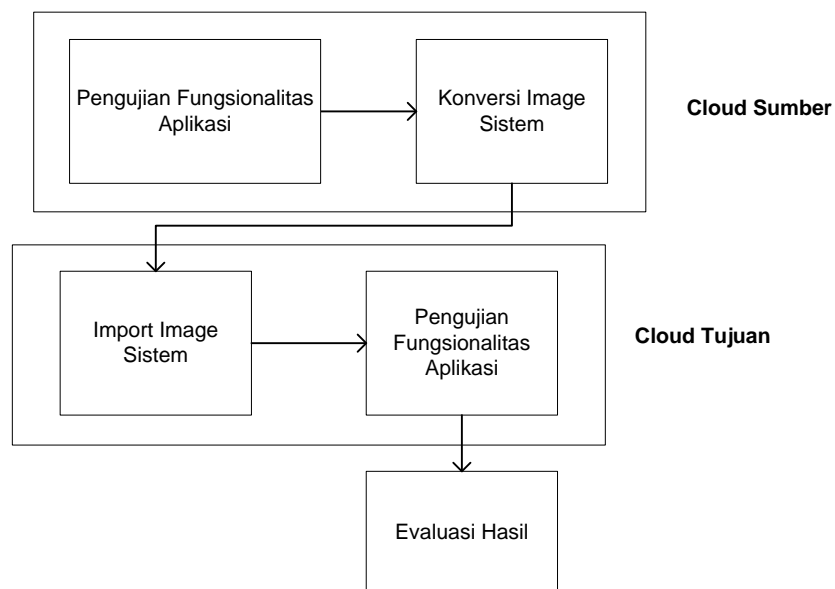
Halaman ini sengaja dikosongkan

BAB 3

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang akan dilakukan pada penelitian ini merupakan metode pengembangan dari metode TIOSA (Lenk dkk, 2014), dimana terdapat penambahan komponen pengujian yaitu pengujian fungsionalitas aplikasi sebelum dan sesudah proses migrasi.



Gambar 3.1 Metode Penelitian

Tahapan metode dari penelitian yang akan dilakukan yaitu:

1. Pengujian Fungsionalitas Aplikasi di Cloud Sumber

Sebelum dilakukan proses migrasi, pengujian fungsionalitas aplikasi dilakukan untuk memastikan fungsionalitas berjalan dengan baik sebelum dilakukan proses migrasi. Pengujian fungsionalitas aplikasi di *cloud* sumber ini yang akan dijadikan sebagai acuan sebagai untuk pengujian fungsionalitas selanjutnya yaitu pada saat setelah migrasi.

2. Konversi Image Sistem ke Cloud Tujuan

Pada tahap ini dilakukan pencarian informasi dan studi mengenai kemungkinan migrasi antara dua *cloud provider* yang akan dilakukan proses migrasi seperti adanya *tools* untuk melakukan konversi *image* dari *cloud* sumber ke *cloud* tujuan kemudian pencarian informasi mengenai standar *image* sistem yang digunakan di kedua *cloud provider*. Proses konversi *image* sistem dilakukan menggunakan *tools* yang tersedia yang disesuaikan dengan jenis *cloud provider* sumber dan tujuan. Proses konversi *image* sistem dapat bersifat otomatis dan manual.

3. Import Image Sistem di Cloud Tujuan

Proses impor sistem dilakukan setelah seluruh konfigurasi di *cloud* tujuan telah siap untuk dilakukan proses impor *image* dari *cloud* sumber. Proses impor *image* dalam tahap ini dapat bersifat otomatis atau manual sesuai dengan jenis *cloud* sumber dan *cloud* tujuan yang digunakan.

4. Menguji aplikasi

Setelah tahap impor *image* sistem berhasil dilakukan dan seluruh layanan dalam sistem yang telah dimigrasi mulai dijalankan termasuk aplikasi yang akan dilakukan pengujian fungsionalitasnya.

Pengujian fungsionalitas aplikasi pada *cloud* tujuan menggunakan *testbed* yang memiliki komponen pengujian yang sama dengan pengujian di server fisik. Tujuannya yaitu melakukan verifikasi fungsionalitas aplikasi.

5. Mengevaluasi hasil

Setelah seluruh data pengujian migrasi terkumpul maka akan dapat ditentukan tingkat interoperabilitas sesuai dengan parameter ukuran yang telah ditentukan. Untuk setiap kemungkinan migrasi akan digolongkan sesuai dengan tingkat interoperabilitas aplikasinya.

3.2 Rancangan Sistem

Rancangan sistem yang akan dijelaskan dalam bab ini terdiri dari 2 yaitu, rancangan desain infrastruktur pengujian sistem dan rancangan pengujian aplikasi sistem yang akan digunakan sebagai uji fungsionalitas antara 2 *cloud* yang berbeda.

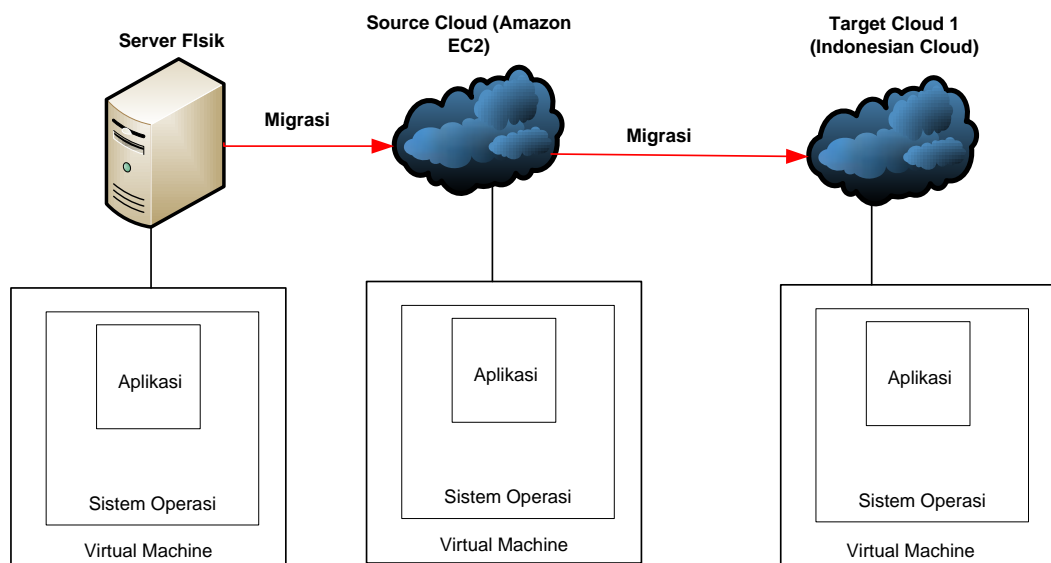
3.2.1 Skenario Desain Migrasi Sistem

Terdapat 2 macam infrastruktur *cloud* yang akan dilakukan pengujian tingkat interoperabilitasnya. Infrastruktur pertama yaitu migrasi *cloud* antara Amazon Elastic Compute Cloud (EC2) dimana jenis *hypervisor* yang dipakai yaitu Xen dengan Indonesian Cloud dimana *hypervisor* yang dipakai yaitu VMware vCloud Air. Sedangkan untuk infrastruktur migrasi *cloud* yang kedua yaitu antara Amazon EC2 (Elastic Compute Cloud) dimana jenis *hypervisor* yang dipakai yaitu Xen dengan Google Compute Engine yang memakai KVM sebagai *hypervisor*nya. Skenario migrasi yang dilakukan terbagi menjadi 2 skenario yaitu:

1. Migrasi Amazon EC2 ke Indonesian Cloud

Pada infrastruktur migrasi *cloud* dari Amazon EC2 ke Indonesian Cloud tahapan yang dilakukan yaitu:

- Melakukan pengujian fungsionalitas aplikasi di server fisik
- Migrasi image dari server fisik ke server Amazon EC2
- Melakukan pengujian fungsionalitas aplikasi
- Migrasi image dari server Amazon EC2 ke Indonesian Cloud
- Melakukan pengujian fungsionalitas aplikasi di server Indonesian Cloud

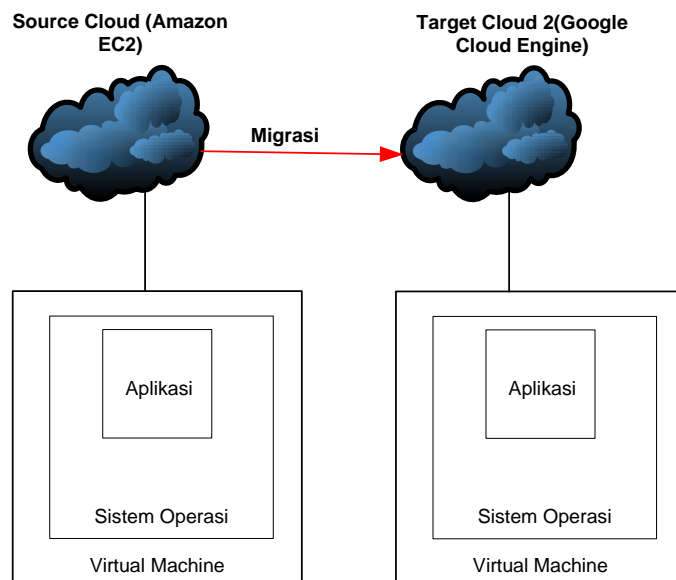


Gambar 3.2 Skenario Migrasi Pertama

2. Migrasi Amazon EC2 ke Google Compute Engine

Pada infrastruktur migrasi *cloud* dari Amazon EC2 ke Google Compute Engine tahapan yang dilakukan yaitu :

- a. Melakukan pengujian fungsionalitas aplikasi di server Amzon EC2
- b. Migrasi image dari server Amazon EC2 ke Google Compute Engine
- c. Melakukan pengujian fungsionalitas aplikasi di server Google Compute Engine



Gambar 3.3 Skenario Migrasi Kedua

3.2.2 Aplikasi E-Commerce

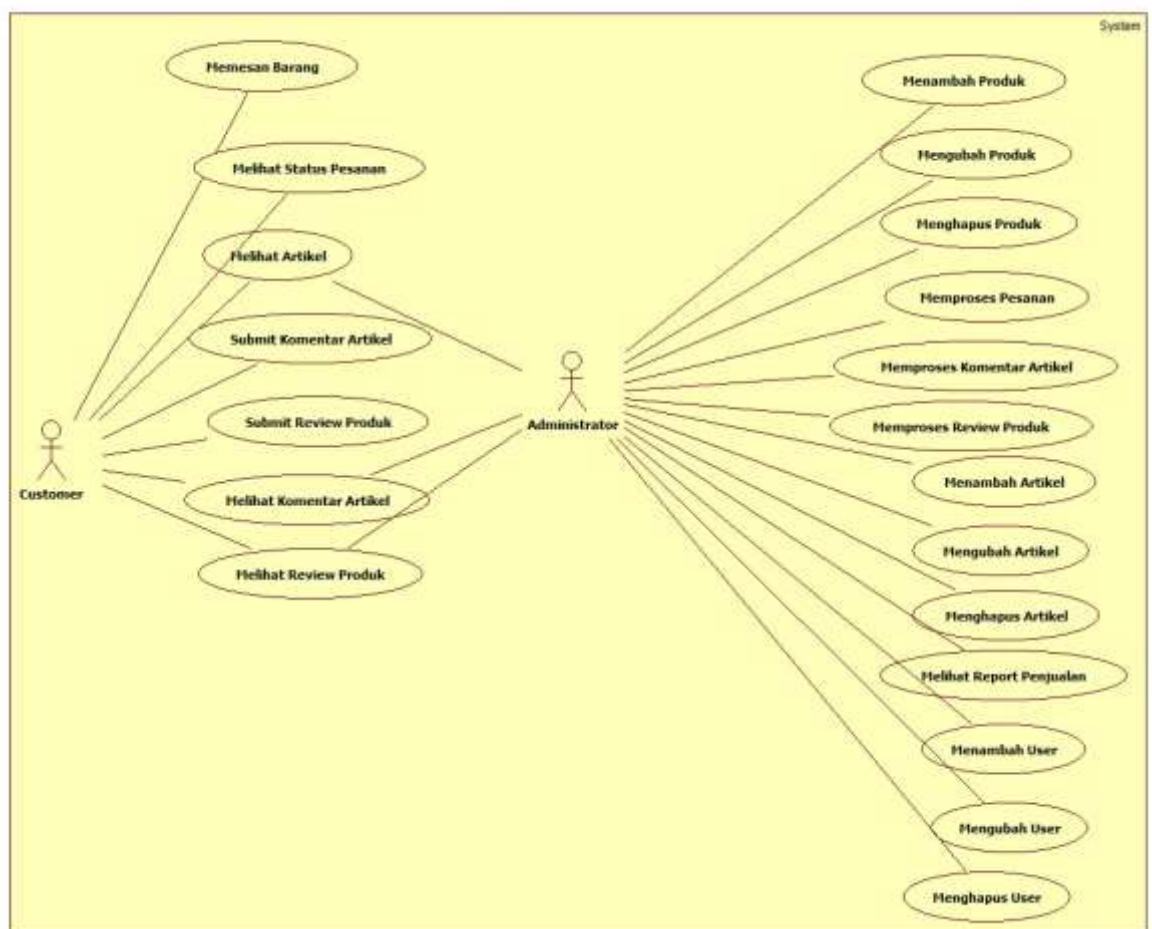
Aplikasi yang akan diuji tingkat interoperabilitasnya dalam penelitian ini yaitu layanan E-Commerce. E-Commerce merupakan *website* yang menyediakan fasilitas membeli dan menjual produk melalui *internet*. Terdapat berbagai macam fitur layanan dalam suatu *website* E-Commerce mulai dari pencarian produk yang diinginkan pembeli, pemesanan produk, hingga transaksi.

Dalam penelitian ini aplikasi yang digunakan dalam melakukan pengujian interoperabilitas yaitu aplikasi E-Commerce berbasis *content manegement system* yang disediakan oleh *wordpress* menggunakan plugin *woocommerce*. Diagram

Use Case dan Diagram Activity dalam penelitian ini digunakan untuk mempermudah dalam menentukan fungsionalitas aplikasi apa saja yang akan diuji.

1. Diagram Use Case Aplikasi

Diagram *use case* pada aplikasi menggambarkan menggambarkan fungsi-fungsi yang terdapat pada sebuah ^{sistem} dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.



Gambar 3.4 Diagram Use Case Aplikasi Pengujian

Dalam sistem aplikasi yang akan diuji terdapat 2 aktor yaitu Customer dan Administrator. Customer dalam aplikasi merupakan pengguna dari aplikasi E-Commerce yang bisa melakukan berbagai macam fungsi, mulai dari melakukan pemesanan barang, melihat status pemesanan barang yang telah dipesan, melihat

artikel yang dibuat oleh administrator, melakukan submit komentar pada artikel, melakukan submit review pada produk, melihat komentar artikel, serta melihat review produk.

Sedangkan untuk aktor Administrator merupakan pengelola dari aplikasi E-Commerce yang mempunyai beberapa fungsi yaitu menambah produk, mengubah produk, menghapus produk, memproses pesanan, memproses komentar artikel, memproses review produk, menambah artikel, mengubah artikel, menghapus artikel, melihat report penjualan, menambah, mengubah user, serta menghapus user.

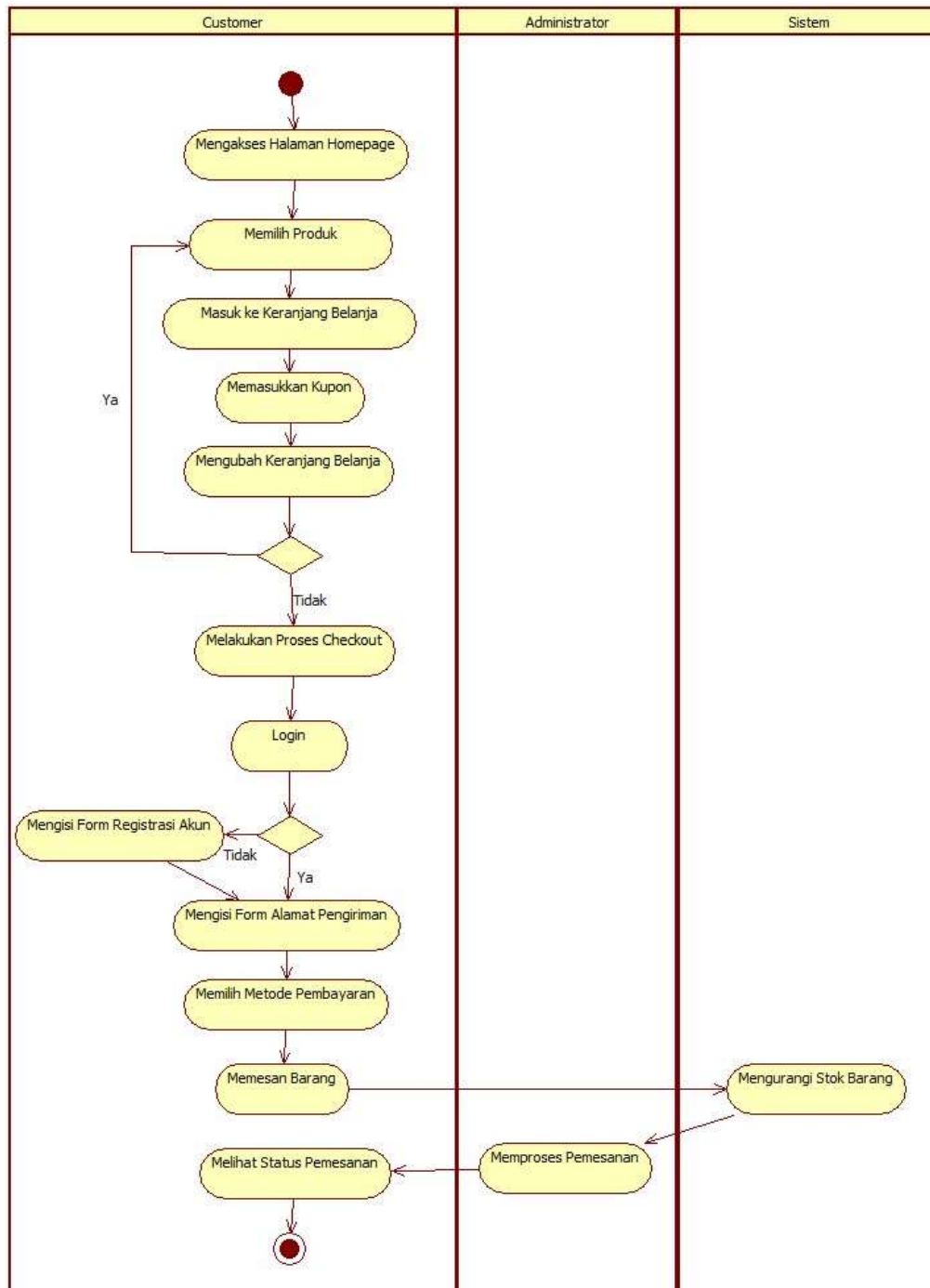
2. Diagram Activity Aplikasi

Diagram activity menggambarkan alur proses yang terjadi pada aplikasi. Pada penelitian ini digunakan diagram *activity* /aktifitas untuk menggambarkan alur proses yang terjadi pada pada tiap modul yang akan diuji. Adapun beberapa flow aktifitas yang akan diuji yaitu:

a. Memesan Barang

Alur proses yang terjadi pada saat customer memesan barang yaitu customer mengakses halaman homepage toko, kemudian memilih produk yang akan dipesan kemudian halaman akan mengarahkan customer ke keranjang belanja setelah itu customer memasukkan kupon yang dimiliki. Jika customer ingin mengubah keranjang belanja maka customer dapat kembali memilih produk yang akan ditambah maupun dikurangi. Jika tidak maka customer melakukan proses checkout. Dalam proses checkout customer diharuskan untuk login terlebih dahulu, kemudian mengisi data alamat pengiriman dan memilih metode pembayaran. Setelah itu customer baru bisa melakukan proses pemesanan. Jika customer tidak memiliki user login maka sebelum mengisi form alamat pembayaran maka customer harus melakukan registrasi akun terlebih dahulu. Setelah proses pemesanan berhasil maka sistem otomatis akan mengurangi jumlah stok barang sesuai dengan jumlah pesanan customer. Administrator berperan sebagai pengelola yang bertugas memproses pemesanan barang dari

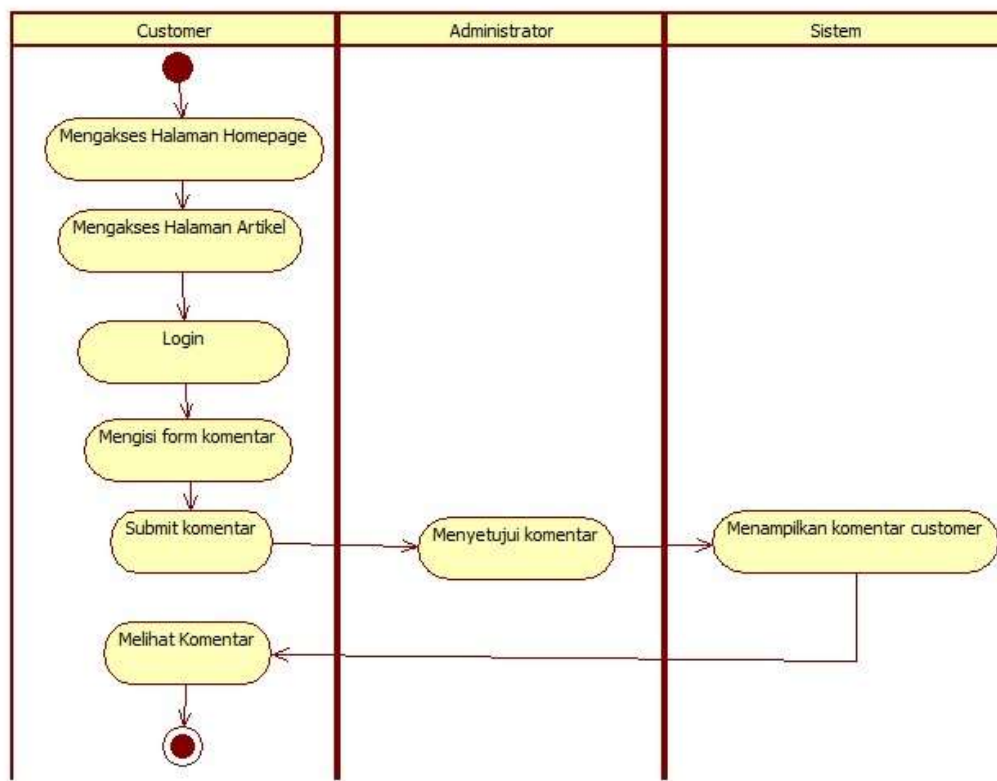
customer. Setelah proses pemesanan berhasil customer juga dapat melihat status pemesanan untuk mengetahui status dari pemesanan yang telah dilakukan.



Gambar 3.5 Diagram Aktifitas Mesanan Barang

b. Menambah Komentar Artikel

Alur sistem pada aktifitas penambahan komentar artikel yaitu setelah customer mengakses halaman toko, lalu mengakses ke halaman artikel. customer harus melakukan login terlebih dahulu untuk dapat mengisi komentar artikel. Setelah proses submit artikel berhasil, selanjutnya yaitu administrator akan melakukan persetujuan komentar, setelah itu sistem akan menampilkan komentar customer tersebut. Jika komentar ditolak maka komentar akan dibuang dan tidak akan ditampilkan

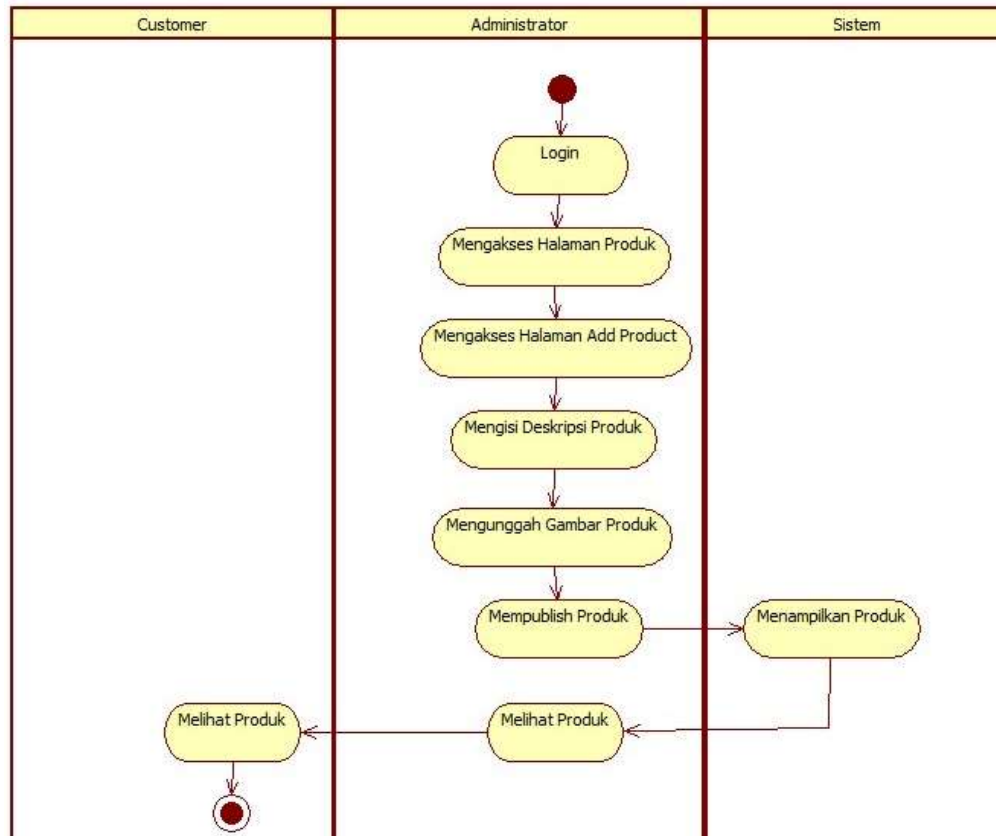


Gambar 3.6 Diagram Aktifitas Menambah Komentar Pada Artikel

c. Menambah Produk

Alur sistem pada aktifitas menambah produk yaitu administrator melakukan login kemudian mengakses halaman produk dan memilih halaman Add Product. Setelah deskripsi produk diisi dan gambar produk terupload, administrator mempublish produk. Jika sukses sistem akan menampilkan produk

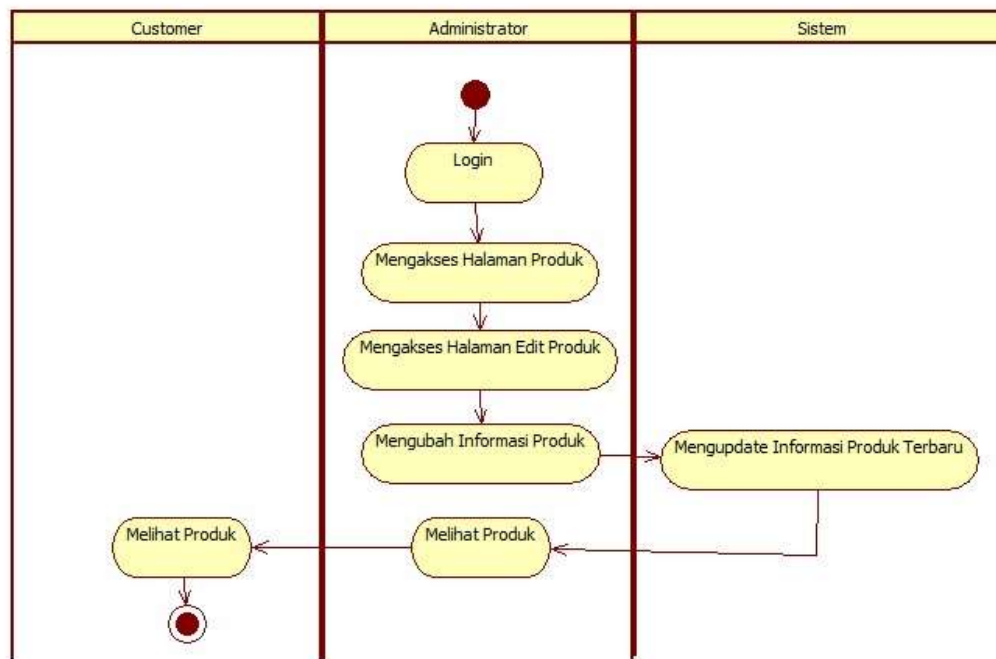
dan administrator, serta customer akan dapat melihat produk baru yang telah dibuat.



Gambar 3.7 Diagram Aktivitas Menambah Produk

d. Mengubah Produk

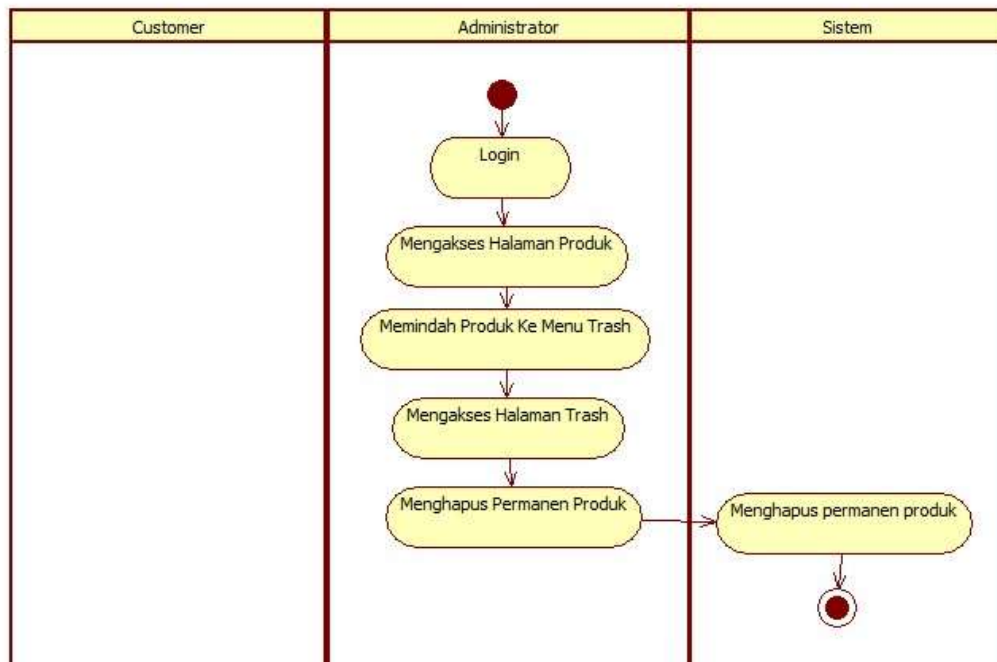
Alur sistem untuk aktifitas mengubah produk yaitu administrator login dan mengakses halaman produk, kemudian memilih produk yang ingin diubah. Setelah informasi produk diubah, sistem akan mengupdate informasi produk terbaru, kemudian administrator dan customer akan dapat melihat informasi produk yang terbaru



Gambar 3.8 Diagram Aktivitas Mengubah Produk

e. Menghapus Produk

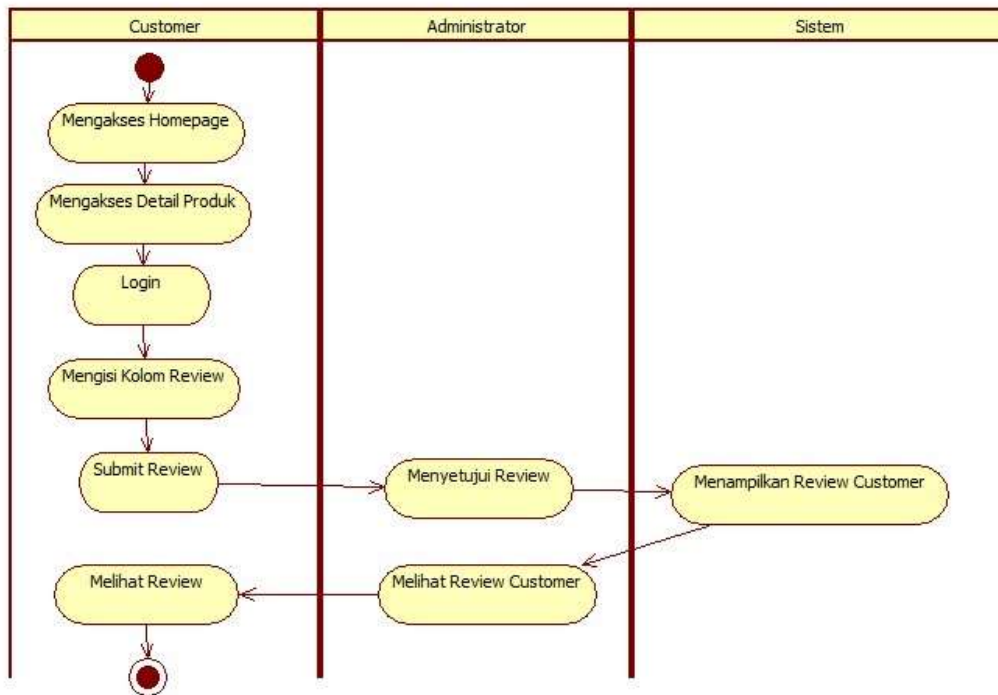
Alur sistem pada saat aktifitas menghapus produk yaitu administrator login terlebih dahulu kemudian mengakses halaman produk, serta memilih produk yang akan dihapus dengan memindahkan produk ke halaman Trash. Setelah mengakses halaman Trash, administrator menghapus permanen produk. Setelah sistem berhasil menghapus produk secara permanen administrator dan customer tidak dapat melihat produk tersebut.



Gambar 3.9 Diagram Aktifitas Menghapus Produk

f. Menambah Review Produk

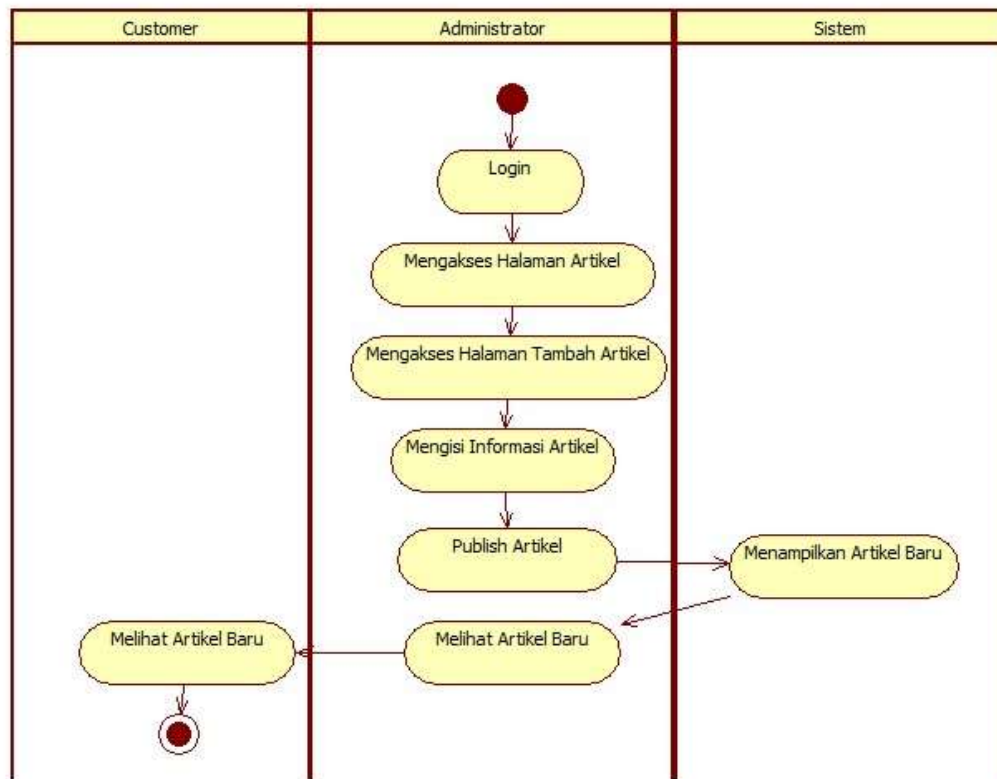
Alur sistem pada aktifitas menambah review produk yaitu customer mengakses halaman homepage website terlebih dahulu, kemudian memilih detail produk. Untuk melakukan submit review produk customer harus melakukan login terlebih dahulu, setelah itu mengisi informasi review produk dan memilih rating produk kemudian melakukan submit review. Administrator akan menyetujui review tersebut, dan sistem akan menyimpan perubahan. Setelah itu administrator dan customer akan dapat melihat review customer.



Gambar 3.10 Diagram Aktifitas Menambah Review Produk

g. Menambah Artikel Baru

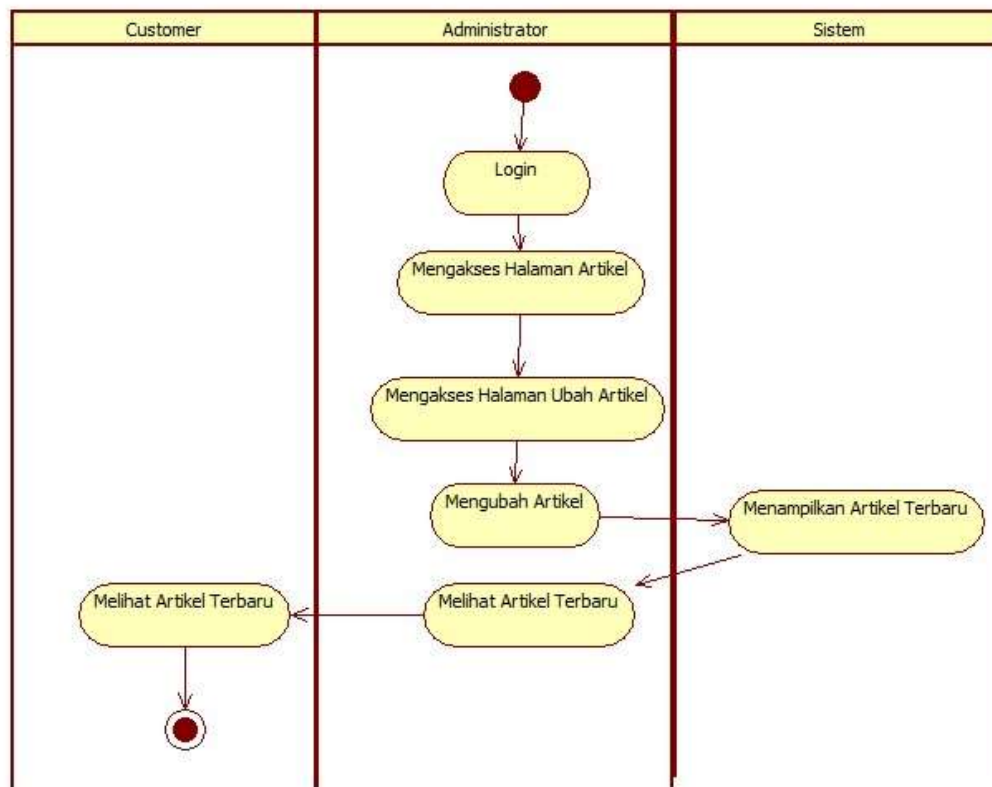
Alur sistem pada aktifitas menambah artikel baru yaitu administrator login dan mengakses halaman artikel, kemudian mengakses halaman tambah artikel baru. Setelah mengisi informasi artikel, administrator melakukan publish artikel dan sistem akan menyimpan artikel baru tersebut. Customer akan dapat melihat artikel baru yang ditambahkan.



Gambar 3.11 Diagram Aktifitas Menambah Artikel Baru

h. Mengubah Artikel

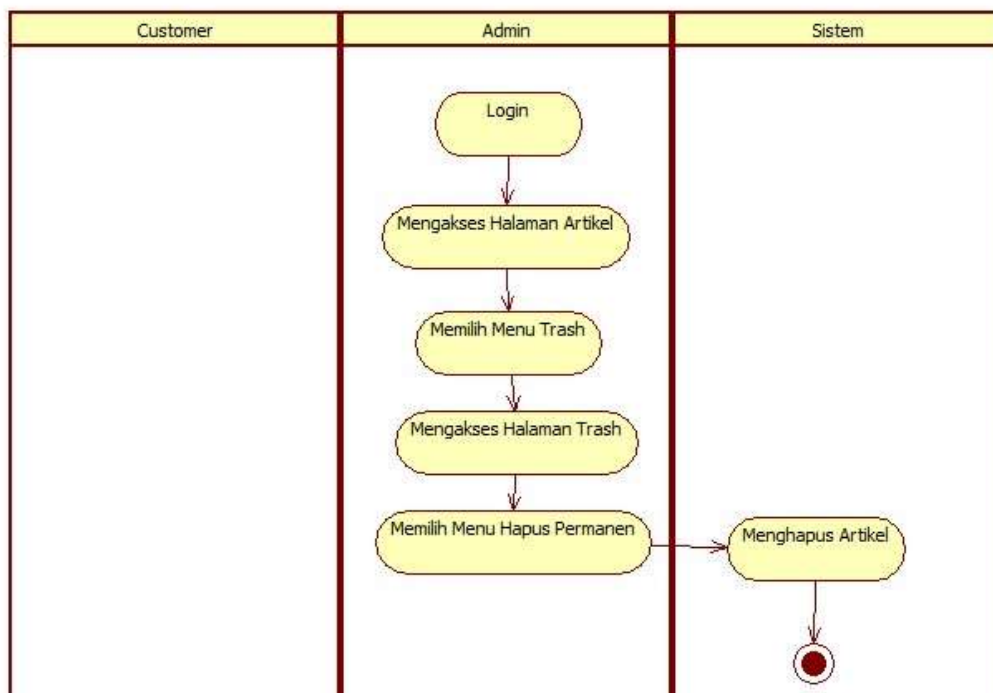
Alur sistem pada aktifitas mengubah artikel yaitu administrator login dan mengakses halaman artikel, kemudian memilih artikel yang akan diubah dengan mengakses halaman ubah artikel. Setelah administrator mengubah isi artikel sistem akan menyimpan perubahan dan selanjutnya administrator serta customer akan dapat melihat isi artikel yang terbaru.



Gambar 3.12 Diagram Aktivitas Mengubah Artikel

i. Menghapus Artikel

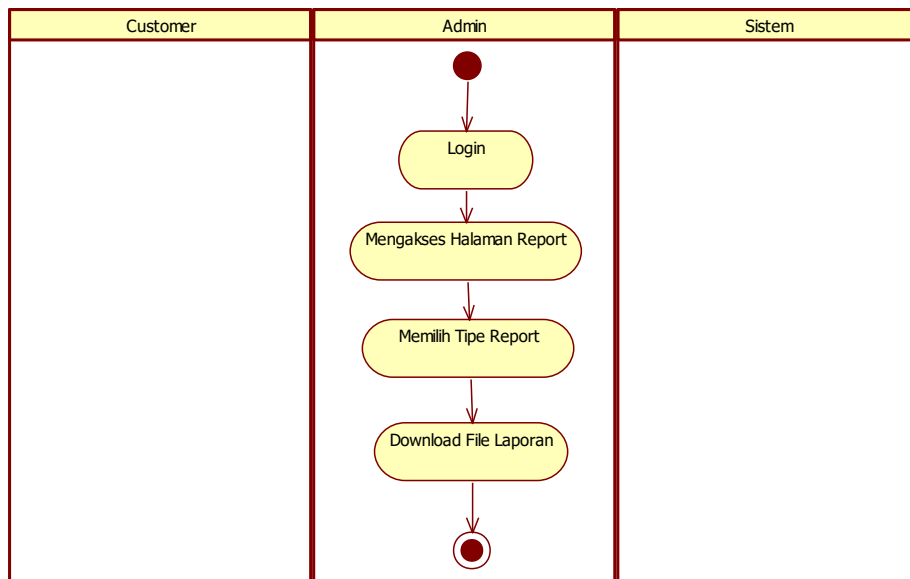
Alur sistem pada aktifitas menghapus artikel yaitu administrator login memilih artikel yang dihapus dengan memilih menu Trash pada artikel yang ingin dihapus. Setelah masuk ke halaman Trash, administrator memilih menu hapus permanen, kemudian sistem akan menghapus artikel tersebut.



Gambar 3.13 Diagram Aktifitas Menghapus Artikel

j. Melihat Report

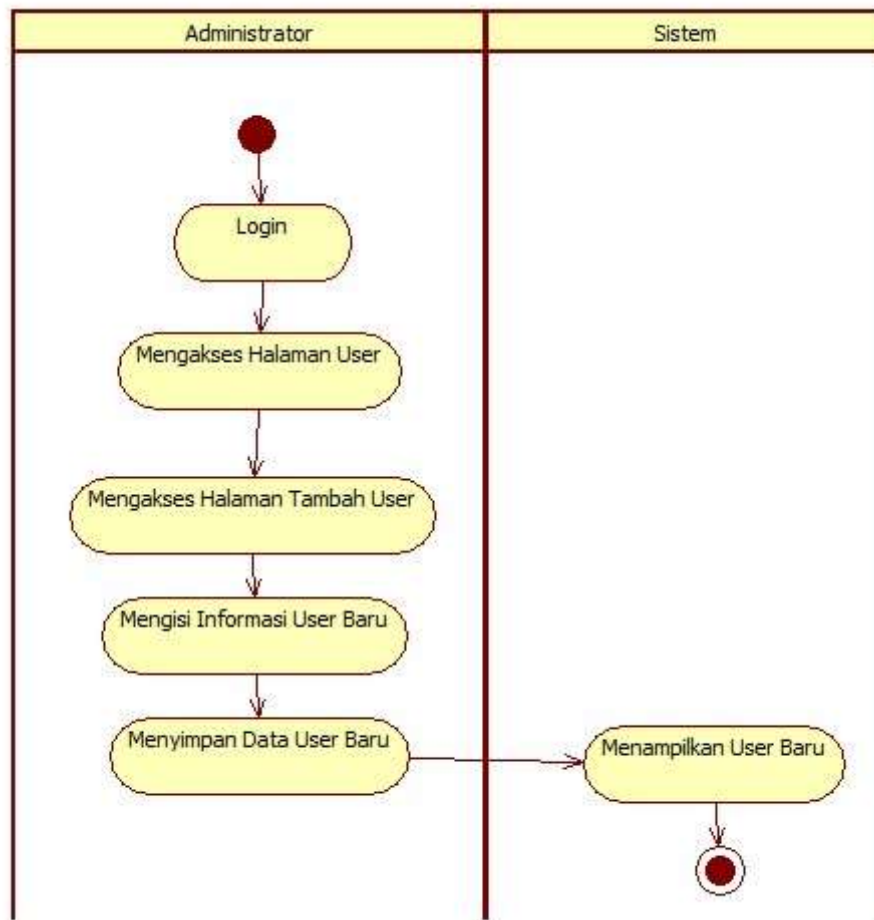
Alur sistem pada aktifitas melihat report yaitu administrator login dan mengakses halaman Report. Setelah itu administrator dapat memilih tipe report yang tersedia sesuai keinginan dan kebutuhan. Administrator juga dapat mendownload file laporan yang telah dipilih tadi.



Gambar 3.14 Diagram Aktivitas Report

k. Menambah User Baru

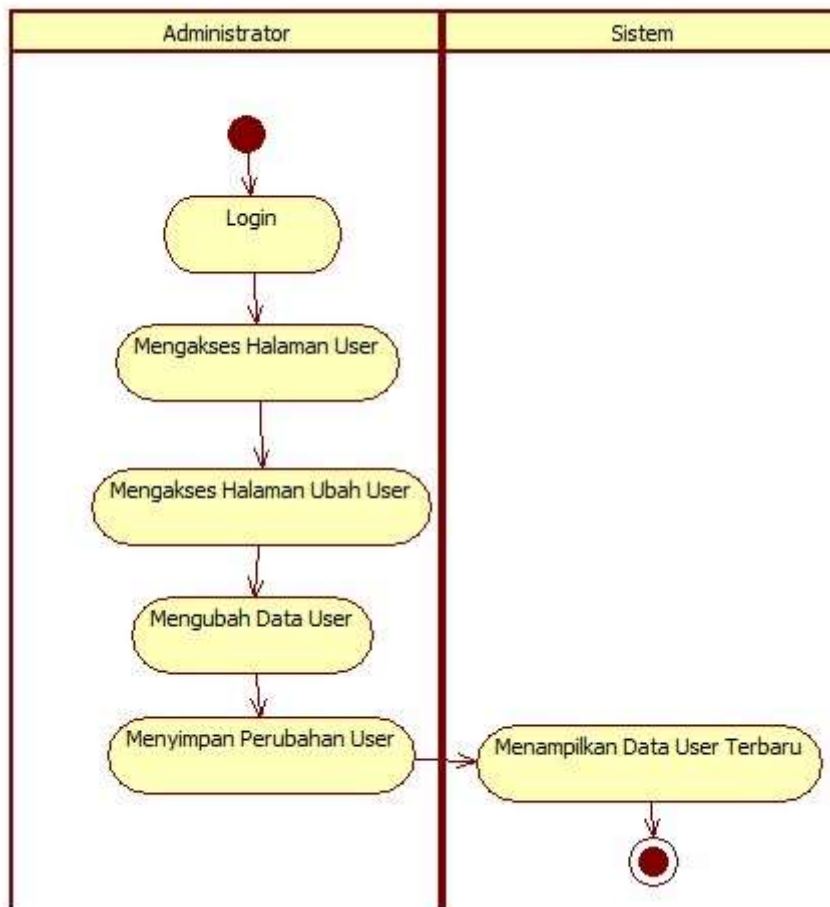
Alur sistem pada aktifitas menambah user baru yaitu administrator login dan mengakses halaman user. Kemudian administrator memilih halaman tambah user baru dan mengisi informasi user baru yang akan dibuat. Setelah itu administrator memilih menu simpan user baru dan sistem akan menyimpan dan menampilkan data user baru yang telah dibuat.



Gambar 3.15 Diagram Aktifitas Menambah User Baru

1. Mengubah Data User

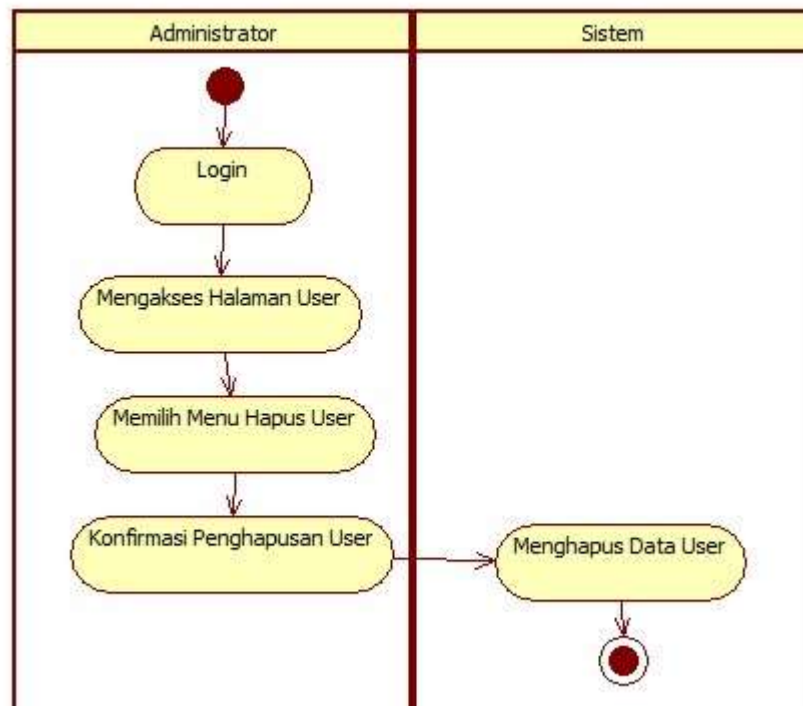
Alur sistem pada aktifitas menambah user baru yaitu administrator login dan mengakses halaman user. Setelah itu administrator memilih user yang akan diubah datanya dengan memilih dan mengakses halaman ubah user. Ketika informasi user telah diisi dengan data terbaru, administrator akan memilih menu simpan perubahan dan sistem akan menampilkan data user terbaru sesuai data yang diubah.



Gambar 3.16 Diagram Aktivitas Mengubah User

m. Menghapus Data User

Alur sistem pada aktifitas menghapus data user yaitu administrator login dan mengakses halaman user. Administrator akan memilih user yang akan dihapus dengan memilih menu hapus user, sistem akan menampilkan konfirmasi penghapusan user. Ketika administrator telah mengkonfirmasi data user yang akan dihapus maka sistem akan menghapus data user tersebut.



Gambar 3.17 Diagram Aktivitas Menghapus User

3.3 Skenario Pengujian

Terdapat beberapa tahap dalam skenario pengujian dalam penelitian ini yaitu:

1. Pengujian Migrasi dari Amazon EC2 ke Indonesian Cloud
 - a. Pengujian fungsionalitas aplikasi di server fisik
 - b. Pengujian ekspor image dari server fisik ke server Amazon EC2
 - c. Pengujian import image di server Amazon EC2
 - d. Pengujian fungsionalitas aplikasi di server Amazon EC2
 - e. Pengujian ekspor image dari server Amazon EC2 ke server Indonesian Cloud
 - f. Pengujian import image di server Indonesian Cloud
 - g. Pengujian fungsionalitas aplikasi di server Indonesian Cloud
2. Pengujian Migrasi dari Amazon EC2 ke Indonesian Cloud
 - a. Pengujian fungsionalitas aplikasi di server Amazon EC2
 - b. Pengujian ekspor image dari server Amazon EC2 ke Google Cloud Engine

- c. Pengujian import image di server Google Cloud Engine
- d. Pengujian fungsionalitas aplikasi di server Google Cloud Engine

3.4 Rancangan Evaluasi Hasil Pengujian

Rancangan hasil evaluasi yang pertama yaitu evaluasi fungsionalitas aplikasi. Rancangan evaluasi pengujian aplikasi ini dirancang berdasarkan alur sistem yang telah disusun pada diagram aktifitas sehingga pengujian dapat dilakukan secara terurut dan memudahkan evaluasi jika terjadi fungsionalitas yang tidak berjalan saat pengujian. Evaluasi ini akan selalu dipakai untuk pengujian fungsionalitas aplikasi baik di *cloud* sumber maupun di *cloud* tujuan setelah proses migrasi berhasil.

Tabel 3.1 Rancangan Testbed Fungsionalitas Aplikasi

No	Aktifitas	Detail Aktifitas	Aktor	Hasil
1	Memesan Barang	1.1 Mengakses Halaman Homepage	Customer	
		1.2 Memilih Produk	Customer	
		1.3 Masuk ke Keranjang Belanja	Customer	
		1.4 Memasukkan Kupon	Customer	
		1.5 Mengubah Keranjang Belanja	Customer	
		1.6 Melakukan Proses Checkout	Customer	
		1.7 Mengisi Form Registrasi Akun	Customer	
		1.8 Mengisi Form Alamat Pengiriman	Customer	
		1.9 Memilih Metode Pembayaran	Customer	
		1.10 Memesan Barang	Customer	
		1.11 Mengurangi Stok Barang	Sistem	
		1.12 Memproses Pemesanan	Administrator	
		1.13 Melihat Status Pemesanan	Customer	
2	Menambah Produk	2.1 Login	Administrator	
		2.2 Mengakses Halaman Produk	Administrator	
		2.3 Mengakses Halaman Add Product	Administrator	
		2.4 Mengisi Deskripsi Produk	Administrator	
		2.5 Mengunggah Gambar Produk	Administrator	
		2.6 Mempublish Produk	Administrator	
		2.7 Menampilkan Produk	Sistem	
		2.9 Melihat Produk Baru	Administrator	

No	Aktifitas	Detail Aktifitas	Aktor	Hasil
		2.10 Melihat Produk Baru	Customer	
3	Mengubah Produk	3.1 Login	Administrator	
		3.2 Mengakses Halaman Produk	Administrator	
		3.3 Mengakses Halaman Edit Produk	Administrator	
		3.4 Mengubah Informasi Produk	Administrator	
		3.6 Mengupdate Informasi Produk Terbaru	Sistem	
		3.5 Melihat Produk	Administrator	
		3.6 Melihat Produk	Customer	
4	Menghapus Produk	4.1 Login	Administrator	
		4.3 Mengakses Halaman Produk	Administrator	
		4.4 Memindah Produk Ke Menu Trash	Administrator	
		4.5 Mengakses Halaman Trash	Administrator	
		4.6 Menghapus Permanen Produk	Administrator	
		4.7 Menghapus Permanen Produk	Sistem	
5	Menambah Komentar Artikel	5.1 Mengakses Halaman Homepage	Customer	
		5.2 Mengakses Halaman Artikel	Customer	
		5.2 Login	Customer	
		5.3 Mengisi form komentar	Customer	
		5.4 Submit komentar	Customer	
		5.5 Menyetujui komentar	Administrator	
		5.6 Menampilkan komentar	Sistem	
		5.7 Menyetujui komentar	Administrator	
		5.8 Melihat Komentar	Customer	
6	Mensubmit Review Produk	6.1 Mengakses Homepage		
		6.2 Mengakses Detail Produk	Customer	
		6.3 Login	Customer	
		6.4 Mengisi Kolom Review	Customer	
		6.5 Submit Review	Customer	
		6.6 Menyetujui Review	Administrator	
		6.8 Menampilkan Review	Sistem	

No	Aktifitas	Detail Aktifitas	Aktor	Hasil
		Customer		
		6.9 Melihat Review Customer	Administrator	
		6.10 Melihat Review	Customer	
7	Menambah Artikel	7.1 Login	Administrator	
		7.2 Mengakses Halaman Artikel	Administrator	
		7.3 Mengakses Halaman Tambah Artikel	Administrator	
		7.4 Mengisi Informasi Artikel	Administrator	
		7.5 Publish Artikel	Administrator	
		7.6 Menampilkan Artikel Baru	Sistem	
		7.7 Melihat Artikel Baru	Administrator	
		7.8 Melihat Artikel Baru	Customer	
8	Mengubah Artikel	8.1 Login	Administrator	
		8.2 Mengakses Halaman Artikel	Administrator	
		8.3 Mengakses Halaman Ubah Artikel	Administrator	
		8.4 Mengubah Artikel	Administrator	
		8.5 Menampilkan Artikel Terbaru	Sistem	
		8.6 Melihat Artikel Terbaru	Administrator	
		8.7 Melihat Artikel Terbaru	Customer	
9	Menghapus Artikel	9.1 Login	Administrator	
		9.2 Mengakses Halaman Artikel	Administrator	
		9.3 Memilih Menu Trash	Administrator	
		9.4 Mengakses Halaman Trash	Administrator	
		9.5 Memilih Menu Hapus Permanen	Administrator	
		9.6 Menghapus Artikel	Sistem	
10	Melihat Report	10.1 Login	Administrator	
		10.2 Mengakses Halaman Report	Administrator	
		10.3 Memilih Tipe Report	Administrator	
		10.4 Download File Laporan	Administrator	
11	Menambah User Baru	11.1 Login	Administrator	
		11.2 Mengakses Halaman User	Administrator	
		11.3 Mengakses Halaman Tambah User	Administrator	
		11.4 Mengisi Informasi User Baru	Administrator	
		11.5 Menyimpan Data User Baru	Administrator	
		11.6 Menampilkan Data User Baru	Sistem	
12	Mengubah User	12.1 Login	Administrator	
		12.2 Mengakses Halaman User	Administrator	

No	Aktifitas	Detail Aktifitas	Aktor	Hasil
		12.3 Mengakses Halaman Ubah User	Administrator	
		12.4 Mengubah Data User	Administrator	
		12.5 Menyimpan Perubahan User	Administrator	
		12.6 Menampilkan Data User Terbaru	Sistem	
13	Menghapus User	13.1 Login	Administrator	
		13.2 Mengakses Halaman User	Administrator	
		13.3 Memilih Menu Hapus User	Administrator	
		13.4 Konfirmasi Penghapusan User	Administrator	
		13.5 Menghapus Data User	Sistem	

Setelah rancangan evaluasi pengujian aplikasi, selanjutnya yaitu rancangan pengujian migrasi yang akan dilakukan. Rancangan ini dirancang sesuai dengan skenario pengujian yang dilakukan yang dibagi menjadi 2 skenario. Jenis pengujian dibagi menjadi 2 sub pengujian yaitu Booting dan Login.

Tabel 3.2 Rancangan Testbed Pengujian Migrasi VM

No	Jenis Pengujian		Hasil
1	Migrasi Server Fisik ke Amazon EC2	Booting	
		Login	
	Migrasi Server Amazon EC2 ke Indonesian Cloud	Booting	
		Login	
2	Migrasi Server Amazon EC2 ke Google Compute Engine	Booting	
		Login	

Booting dan Login merupakan parameter yang dipilih untuk menguji apakah proses import dan eksport yang dilakukan selama migrasi berhasil atau tidak.

3.5 Rancangan Parameter Level Evaluasi Interoperabilitas

Rancangan parameter evaluasi interoperabilitas disusun sebagai hasil akhir yang menentukan level interoperabilitas migrasi. Penentuan level yang digunakan mengacu dari penelitian sebelumnya (Lenk dkk, 2014). Terdapat 2 jenis rancangan evaluasi level interoperabilitas yaitu evaluasi di setiap proses migrasi yang terdiri fungsionalitas yang diuji dan evaluasi fungsionalitas secara keseluruhan yang menentukan tingkat interoperabilitas migrasi *cloud*.

Evaluasi level interoperabilitas ini akan dipakai untuk menguji 2 jenis pengujian yaitu pengujian migrasi *image* VM sistem dan pengujian fungsionalitas aplikasi setelah proses migrasi. Evaluasi fungsionalitas dibagi menjadi 3 level yaitu “Berhasil”, “Peringatan”, dan “Gagal”.

Jika pada pengujian fungsionalitas aplikasi level “Berhasil” berarti jika setelah terjadi migrasi tidak ada perubahan fungsionalitas aplikasi yang diuji. Tidak ada *error/defect software* yang ditemukan baik dari secara mayor maupun minor. Fungsionalitas aplikasi berhasil berjalan di *cloud* baru sesuai dengan fungsionalitas nya di sistem atau *cloud* yang lama. Level “Peringatan” diberikan jika setelah migrasi fungsionalitas yang diuji berjalan sesuai dengan fungsionalitasnya di *cloud* lama, tetapi terjadi beberapa perubahan di fungsionalitas lain seperti fungsionalitas minor. Level “Gagal” diberikan jika setelah terjadi migrasi, fungsionalitas yang diuji tidak berjalan sesuai dengan fungsionalitas asli di *cloud* lama baik fungsionalitas mayor maupun minor.

Defect software bersifat major jika error yang terjadi mempengaruhi fungsionalitas utama aplikasi yang bersifat major/utama seperti fitur dari suatu modul tidak berjalan sesuai proses bisnisnya. Sedangkan bersifat minor, jika error yang terjadi mempengaruhi fungsionalitas utama aplikasi.

Tabel 3.3 Deskripsi Level Interoperabilitas Aplikasi

Level	Aplikasi
Berhasil	Jika setelah migrasi tidak ditemukan <i>error/defect software</i> kategori mayor maupun minor.
Peringatan	<p>Jika setelah migrasi ditemukan <i>error/defect</i> yang bersifat minor seperti :</p> <ol style="list-style-type: none"> 1. <i>Banner website</i> tidak muncul pada <i>homepage</i> 2. Terdapat kesalahan redaksional pada aplikasi
Gagal	<p>Jika setelah terjadi migrasi, fungsionalitas yang diuji tidak berjalan sesuai dengan fungsionalitas asli di <i>cloud</i> lama dan ditemukan <i>defect software</i> dikategorikan sebagai <i>defect</i> mayor.</p> <p>Contoh :</p> <ol style="list-style-type: none"> 1. Customer gagal melakukan pemesanan barang setelah memilih menu submit pemesanan 2. Administrator gagal menyimpan produk baru 3. Administrator tidak bisa login ke dalam aplikasi

Sedangkan untuk pengujian migrasi image VM level “Berhasil” berarti jika setelah terjadi migrasi parameter yang diujikan seperti *booting/reboot* berjalan dengan baik. Level “Peringatan” diberikan jika setelah migrasi parameter yang

diujikan seperti *booting/reboot* berjalan normal tetapi terdapat perubahan di parameter lain yang tidak terlalu signifikan. Level “Gagal” diberikan jika setelah terjadi migrasi, parameter sistem yang diuji tidak berjalan sebagaimana mestinya.

Tabel 3.4 Evaluasi Hasil Keseluruhan

Level	Dekripsi Penilaian
Berhasil	Jika seluruh parameter yang diujikan mendapatkan status level ‘Berhasil’.
Peringatan	Jika dari seluruh parameter yang diuji terdapat minimal 1 parameter yang mendapatkan status ‘Peringatan’
Gagal	Jika dari seluruh parameter yang diuji terdapat 1 parameter yang mendapatkan status ‘Gagal’

Evaluasi hasil keseluruhan dirancang sebagai evaluasi hasil akhir dari seluruh pengujian baik dari migrasi image VM maupun fungsionalitas aplikasi.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Migrasi Amazon EC2 ke Indonesian Cloud

Proses migrasi dari Amazon EC2 ke Indonesian Cloud dilakukan melalui beberapa tahapan yaitu:

1. Ekspor image VM (Virtual Machine) sistem fisik dalam format OVA

Server fisik yang digunakan dalam penelitian ini menggunakan *hypervisor* berjenis VMWare VSphere dimana menyediakan fasilitas ekspor image VM dalam bentuk format OVA. Format OVA merupakan salah satu format yang didukung oleh Open Virtualization Framework(OVF) dimana merupakan standar pembangunan Virtual Machine yang didukung oleh DMTF. Standar OVA merupakan standar untuk interoperabilitas VM antar *cloud* yang berbeda infrastruktur *hypervisor*nya. Amazon EC2 merupakan salah satu *cloud provider* yang mendukung format OVA dalam pembuatan *custom image* untuk sistem VM Amazon EC2. Format proses ekspor VM image server fisik ini dilakukan secara otomatis memakai tools yang telah disediakan oleh VMware VSphere.

2. Impor VM image di server Amazon EC2

Setelah proses ekspor berhasil yang ditandai dengan terbentuknya *single image file* dalam format OVA. Proses selanjutnya yaitu melakukan impor VM image tersebut ke dalam server Amazon S3. Amazon S3 merupakan salah satu layanan yang disediakan oleh Amazon Cloud dalam bentuk *storage* atau tempat penyimpanan. Setelah file berhasil terupload pada Amazon S3, proses selanjutnya yaitu melakukan proses import melalui CLI (command line interface) dengan menggunakan perintah import VM yang telah disediakan oleh Amazon EC2. Contoh perintah impor yang digunakan yaitu:

```
aws ec2 import-image --cli-input-json "{ \"Description\":  
  \"Server Lokal OVA\", \"DiskContainers\": [ { \"Description\":  
  \"Server Lokal OVA\", \"UserBucket\": { \"S3Bucket\":  
  \"soffaimport\", \"S3Key\" : \"server-ubuntu-lokal.ova\" } }  
  ] } }
```

`aws ec2 import-image` : merupakan perintah untuk melakukan impor image ke dalam bentuk AMI(Amazon Machine Image)

`--cli-input-json` : merupakan operasi argumen berbasis JSON

`UserBucket` : merupakan informasi penyimpanan pada Amazon S3

`soffaimport` : merupakan nama bucket yang digunakan untuk menyimpan file image

`server-ubuntu-lokal.ova` : merupakan file image yang akan dilakukan proses konversi ke bentuk AMI

Ketika proses impor berhasil, VM import tersebut akan diubah oleh sistem Amazon EC2 menjadi Amazon Machine Image(AMI). AMI merupakan komponen utama pembentukan Virtual Machine pada sistem Amazon EC2. Setiap AMI berisi *template* atau cetak biru dari sebuah Virtual Machine yang berisi informasi *root volume* dari sebuah *image* seperti sistem operasi dan aplikasi yang digunakan, konfigurasi keamanan, serta pemetaan *block device*.

Pembuatan VM baru dalam server Amazon EC2 memakai AMI hasil import server fisik tersebut. Saat pembuatan VM baru, pengguna dapat menspesifikasikan ulang konfigurasi *hardware* dan keamanan yang akan digunakan menyesuaikan dengan pilihan yang tersedia dalam Amazon EC2. Proses akhir migrasi image VM ini diakhiri dengan menghidupkan VM baru yang telah dibangun menggunakan *image* VM dari server fisik tadi, jika proses *booting* dan *startup* berhasil dan aplikasi di dalam sistem berjalan sesuai dengan kondisi di server fisik maka proses migrasi sistem dari server fisik ke server Amazon EC2 telah berhasil.

3. Ekspor image dari server Amazon EC2 ke server Indonesian Cloud

Proses migrasi dari Amazon EC2 ke server Indonesian Cloud dilakukan dengan menggunakan command yang telah disediakan oleh Amazon EC2 dengan menyesuaikan format yang didukung oleh server Indonesian Cloud. Indonesian Cloud merupakan penyedia layanan *cloud* yang berbasis VMware VCloud Air dimana mendukung format OVA jika ingin melakukan proses import VM pada

sistemnya. Hasil dari proses ekspor ini menghasilkan single image berformat OVA. Contoh perintah ekspor yang digunakan yaitu:

```
ec2-create-instance-export-task i-034b43c4 -e VMware -f VMDK -c  
OVA -b soffatesis
```

Keterangan :

`ec2-create-instance-export-task` : merupakan perintah untuk melakukan ekspor VM ke Amazon S3

`i-034b43c4` : merupakan ID dari VM yang akan dilakukan proses ekspor

`-e VMware` : merupakan target dari ekspor image

`-f VMDK` : merupakan file format dari virtual disk

`-c OVA` : merupakan file format dari image sistem yang akan diekspor

`-b soffatesis` : merupakan bucket tujuan ketika file image berhasil diekspor

4. Impor image di server Indonesian Cloud

Seperti pada tahap impor sebelumnya yaitu dari *server* fisik ke Amazon EC2, dalam proses impor ke *server* Indonesian Cloud ini juga diharuskan *upload image* yang telah berhasil diekspor di server Amazon EC2 ke server Indonesian Cloud. Setelah berhasil diupload *image* VM ini akan menjadi template untuk pembuatan VM baru pada *server* Indonesian Cloud.

Pada pembuatan VM baru pada server Indonesian Cloud ini pengguna juga dapat mengatur ulang konfigurasi VM yang digunakan seperti *hardware* dan keamanannya. Proses akhir migrasi image VM ini diakhiri dengan menghidupkan VM baru yang telah dibangun menggunakan image VM dari server Amazon EC2, jika proses booting dan startup berhasil dan aplikasi di dalam sistem berjalan sesuai dengan kondisi di *server* fisik maka proses migrasi sistem dari *server* Amazon EC2 ke *server* Indonesian Cloud telah berhasil.

Tabel 4.1 Hasil Pengujian Migrasi Image VM Amazon EC2 ke Indonesian Cloud

No	Jenis Pengujian		Hasil
1	Migrasi Server Fisik ke Amazon EC2	Booting	Berhasil
		Login	Berhasil
2	Migrasi Server Amazon EC2 ke Indonesian Cloud	Booting	Berhasil
		Login	Berhasil

Hasil pengujian menunjukkan bahwa untuk seluruh pengujian migrasi VM dari server Amazon EC2 ke Indonesian Cloud telah berhasil ditunjukkan dengan berhasilnya proses booting dan login ke VM setelah proses migrasi. Berdasarkan hasil pengujian tersebut perubahan konfigurasi sistem baik dari segi hardware, security yang dilakukan selama migrasi tidak mempengaruhi keberhasilan migrasi VM.

Sedangkan untuk pengujian fungsionalitas aplikasi setelah proses migrasi aplikasi mulai dari server fisik ke server Amazon EC2, kemudian dilanjutkan dari Amazon EC2 ke *server* Indonesian Cloud terdapat *error/defect software* yang termasuk ke dalam klasifikasi *defect* minor yaitu ketika customer mengakses halaman *homepage*, *banner website* tidak muncul.

Tabel 4.2 Hasil Pengujian Fungsionalitas Aplikasi dari Amazon EC2 ke Indonesian Cloud

No	Aktifitas	Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Indonesian Cloud	
		Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
1	Memesan Barang	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Banner website tidak muncul pada homepage aplikasi.
2	Menambah Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
3	Mengubah Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Indonesian Cloud	
		Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
4	Menghapus Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
5	Menambah Komentar Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Banner website tidak muncul pada homepage aplikasi.
6	Mensubmit Review Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Banner website tidak muncul pada homepage aplikasi.
7	Menambah Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
8	Mengubah Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
9	Menghapus Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
10	Melihat Report	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
11	Menambah User Baru	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
12	Mengubah User	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
13	Menghapus User	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

Berdasarkan data hasil pengujian migrasi *image* VM dari *server* fisik ke Amazon EC2 dilanjutkan ke *server* Indonesian Cloud proses migrasi yang

dilakukan berhasil. Alasan mengapa skenario pertama ini harus dimulai dari server fisik terlebih dahulu yaitu migrasi VM dari Amazon EC2 ke Indonesian Cloud hanya dapat dilakukan melalui cara VM Export (Amazon Web Services, 2016). dimana VM image yang bisa diekspor dari *server* Amazon EC2 hanya terbatas pada VM yang sebelumnya juga harus diimpor menggunakan fasilitas VM Import dari Amazon EC2 (Amazon Web Services, 2016) sehingga pada skenario pertama ini harus ditambahkan juga migrasi dari *server* fisik untuk dapat melakukan migrasi dari *server* Amazon EC2 ke *server* Indonesian Cloud.

Saat proses *bundling image* VM pada *server* fisik, format *image* yang digunakan yaitu OVA dimana format file ini merupakan salah satu format yang didukung oleh Amazon EC2 untuk fasilitas import VM *image*. Beberapa format lain yang didukung oleh Amazon EC2 yaitu VMDK, VHD, dan RAW (Amazon Web Services, 2016). Setelah VM image hasil *bundle* di *server* fisik berhasil diupload di *server* Amazon S3 milik Amazon Web Service dan siap untuk dilakukan *proses* import ke Amazon EC2 kendala yang dapat muncul yaitu pembuatan role untuk melakukan download image dari Amazon S3 dimana nantinya akan ditransfer ke Amazon EC2 untuk di konversi menjadi Amazon Machine Image (AMI).

Ketika AMI hasil import sudah terbentuk dan AMI siap untuk diubah menjadi VM baru di *server* Amazon EC2, konfigurasi VM asli seperti konfigurasi *hardware* seperti *memory*, *hardisk*, CPU dapat berubah menyesuaikan dengan konfigurasi pilihan yang telah disediakan oleh Amazon EC2, berdasarkan pengujian perubahan konfigurasi *hardware* ini hanya berpengaruh pada performansi aplikasi saja, belum ditemukan kasus untuk fungsionalitas aplikasi dalam penelitian ini. Seperti pada 4.1 dibawah ini, terlihat bahwa saat sistem masih di *server* fisik, besar konfigurasi *memory* yaitu 2048 MB.



Gambar 4.1 Konfigurasi Hardware Server Fisik

Sedangkan terlihat pada Gambar 4.3 konfigurasi pembuatan VM baru dari image server fisik mengalami perubahan besar memory menjadi 1GiB.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virt and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applic

Filter by: All instance types Current generation Show/Hide Columns

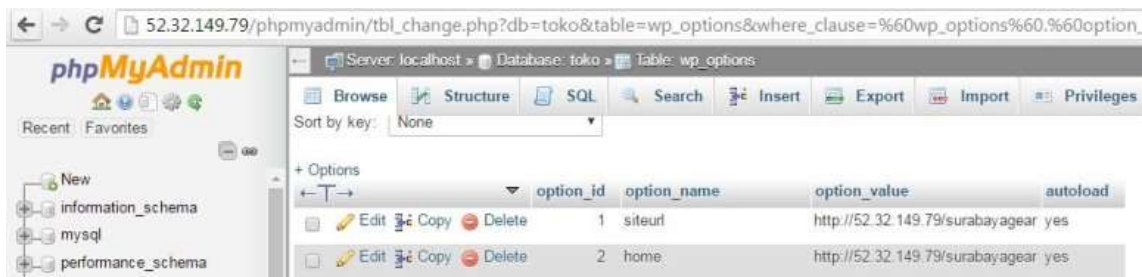
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS onl)

	Family	Type	vCPUs	Memory (GiB)
<input type="checkbox"/>	General purpose	t2.nano	1	0.5
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1
<input type="checkbox"/>	General purpose	t2.small	1	2
<input type="checkbox"/>	General purpose	t2.medium	2	4
<input type="checkbox"/>	General purpose	t2.large	2	8
<input type="checkbox"/>	General purpose	m4.large	2	8

Gambar 4.2 Konfigurasi Hardware Server Amazon EC2

Proses pengujian fungsionalitas aplikasi pada server Amazon EC2 dilakukan ketika VM baru yang dibangun berhasil *booting* dan *startup* kemudian

berhasil menjalankan layanan *web* aplikasi yang telah ada. Aplikasi yang diuji fungsionalitas dalam penelitian ini yaitu bertipe website E-Commerce menggunakan wordpress. Aplikasi yang diuji setelah proses migrasi di server Amazon EC2 juga harus dikonfigurasi ulang menyesuaikan dengan sistem Amazon EC2 seperti alamat IP yang digunakan seperti yang terlihat pada Gambar 4.3. Dalam hal ini IP Address yang digunakan menyesuaikan dengan IP Address pada server Amazon EC2.



Gambar 4.3 Perubahan Konfigurasi Aplikasi

Pada Gambar 4.4 terlihat bahwa akses aplikasi telah berhasil setelah migrasi dari server fisik, dan VM telah berjalan di server Amazon EC2.

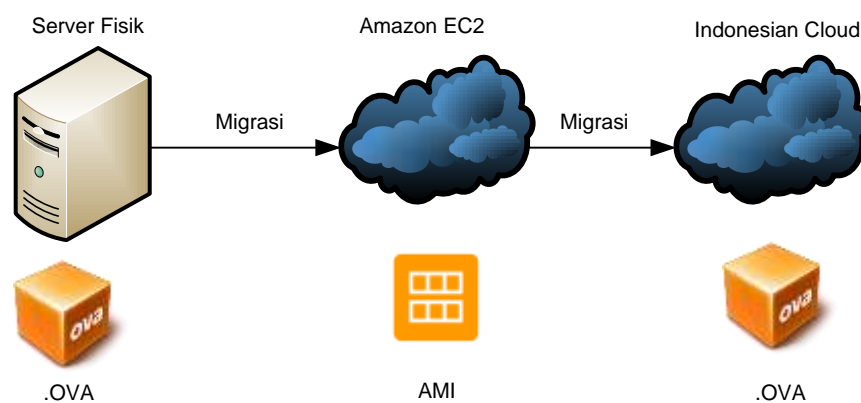


Gambar 4.4 Akses Aplikasi Setelah Migrasi Dari Server Fisik

Setelah dilakukan pengujian fungsionalitas aplikasi menggunakan serangkaian test case dan dibandingkan dengan pengujian fungsionalitas aplikasi di *server* fisik hasil yang didapat yaitu sama dimana semua fungsionalitas aplikasi tidak ada yang berubah setelah dilakukan migrasi dari *server* fisik ke *server* Amazon EC2.

Proses migrasi dari *server* Amazon EC2 ke *server* Indonesian Cloud dilakukan dengan melakukan fasilitas VM Export yang telah disediakan oleh Amazon EC2 yaitu menggunakan *command* `ec2-create-instance-export-task` dengan parameter tipe *file image* yang didukung oleh Indonesian Cloud yaitu OVA yang kemudian setelah berhasil akan disimpan di *server* Amazon S3.

Untuk melakukan proses import ke *server* Indonesian Cloud, tahap yang dilakukan tidak terlalu jauh berbeda dengan proses import image di Amazon EC2 yaitu mulai dari *upload file image* ke *server* Indonesian Cloud lalu pembuatan VM baru di *server* Indonesian Cloud kemudian VM melakukan proses *booting* sampai menjalankan layanan aplikasi E-Commerce sehingga proses migrasi VM *image* dari *server* Amazon EC2 ke *server* Indonesian Cloud yang dilakukan berhasil.



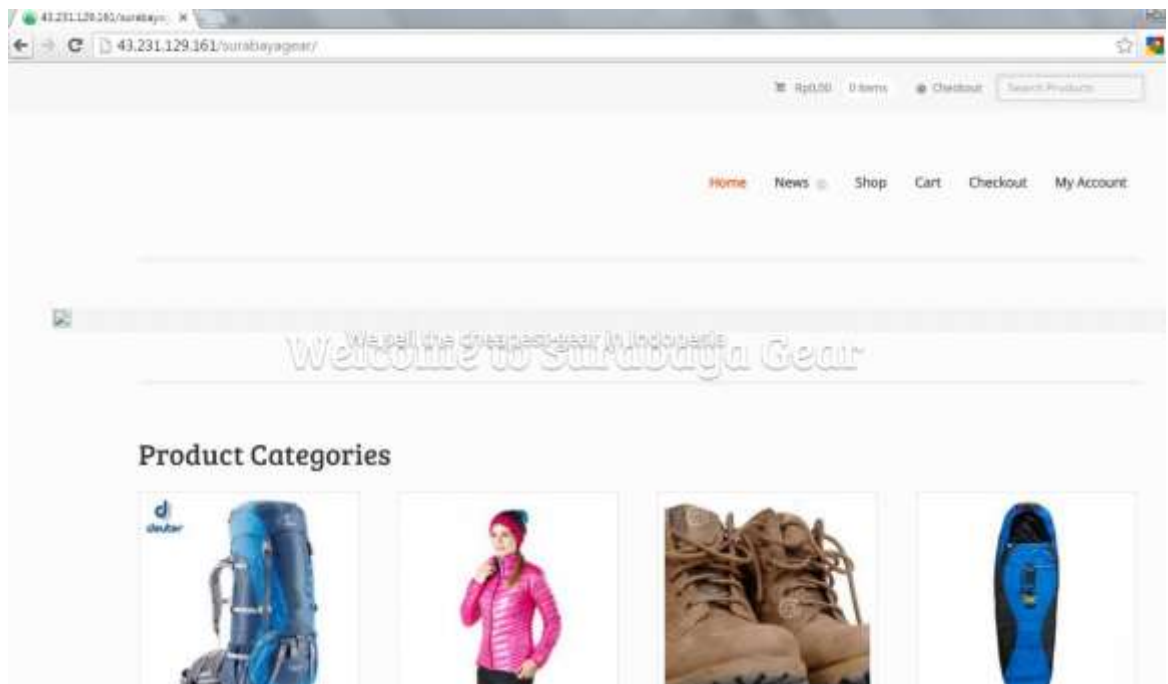
Gambar 4.5 Proses Perubahan Image VM dari Server Fisik ke Server Indonesian Cloud

Proses perubahan *image* selama migrasi menjadi hal yang penting karena menyangkut integritas data yang terdapat pada *image* tersebut, karena terdapat banyak kemungkinan *image* akan mengalami *corrupt* baik saat proses

import/eksport maupun saat *upload* dan *download image* di masing-masing *server*. Format VM image di server fisik yaitu OVA, kemudian ketika dilakukan migrasi ke server Amazon EC2 dilakukan transformasi menjadi format AMI (Amazon Machine Image). Pengujian yang dapat dilakukan yaitu pengujian *booting* dan *login* VM di server Amazon EC2. Pada skenario ini proses *login* berhasil dilakukan menggunakan data *login* yang sama dengan *server* fisik. Kemudian ketika dimigrasikan ke server Indonesian Cloud format *image* berubah menjadi .OVA kembali dan hasil akhirnya sistem dapat berjalan seperti aslinya mulai dari *booting*, *login* dengan *username* yang sama, sampai dengan berhasil diaksesnya layanan *website* E-Commerce oleh *client*.

Pada saat perpindahan migrasi VM dari *server* fisik sampai ke Indonesian Cloud, *credential login* sistem operasi yang dipakai tidak mengalami perubahan selama migrasi. Dalam aspek konfigurasi *security* ketika proses migrasi di Amazon EC2 berhasil, pada saat mengakses/*remote server* di Amazon harus memakai *public & private key* yang telah disediakan oleh Amazon EC2, sedangkan ketika VM telah dimigrasikan ke Indonesian Cloud, *public* dan *private key* Amazon EC2 tidak digunakan kembali. Selain itu konfigurasi IP pada saat setelah migrasi berubah baik di Amazon EC2 maupun di Indonesian Cloud. Jumlah vCPU, besar *memory* dan *hardisk* ketika migrasi ke Amazon EC2 dapat diubah sesuai keinginan, namun ketika migrasi ke Indonesian Cloud besar *hardisk* tidak bisa diubah karena tipe *controller* yang dipakai.

Ketika dilakukan pengujian fungsionalitas aplikasi di server Indonesian Cloud, terdapat fungsionalitas yang berubah yaitu tidak munculnya gambar *banner website* pada *homepage* saat diakses oleh *client*.



Gambar 4.6 Gambar Banner Website Tidak Muncul Setelah Migrasi

Meskipun dari hasil pengujian fungsionalitas aplikasi menggunakan *test case* yang telah dibuat seluruh pengujian *flow* bisnis aplikasi berjalan dengan benar dan berhasil, terdapat *error* atau *software defect* yang bersifat minor yaitu tidak munculnya gambar *website banner*.

Setelah dilakukan analisis kemungkinan penyebab dari munculnya *error* diatas adalah tidak *terupdatenya* link alamat IP dari gambar *banner* tersebut sehingga *website* tidak dapat menampilkan gambar *banner* tersebut padahal untuk gambar yang lain link gambar sudah *terupdate* dan menyesuaikan dengan IP yang baru. Kemungkinan solusi yang dapat dilakukan yaitu melakukan konfigurasi penggantian secara manual alamat IP dari gambar *banner* tersebut.

4.2 Migrasi Amazon EC2 ke Google Compute Engine

Proses migrasi dari Amazon EC2 ke Google Compute Engine dilakukan melalui beberapa tahapan yaitu :

1. Ekspor image dari server server Amazon EC2 ke Google Compute Engine

Seperti pada tahapan migrasi sebelumnya, sebelum migrasi juga harus dilakukan proses pencarian informasi apakah *cloud* tujuan migrasi selanjutnya mendukung migrasi beserta format yang didukung. Dalam migrasi skenario kedua ini *cloud* tujuan merupakan Google Compute Engine dimana menggunakan jenis *hypervisor* berbeda dengan Amazon EC2. Jika Amazon EC2 menggunakan *hypervisor* XEN, Google Compute Engine menggunakan *hypervisor* KVM. Tidak seperti migrasi pada skenario pertama dimana proses ekspor dapat dilakukan dengan cara menggunakan *tools* dan tidak perlu mengkonfigurasi ulang properties VM *image* saat diekspor, pada skenario migrasi kedua ini dilakukan dengan cara manual yaitu mengkonfigurasi sendiri file *grub* dan *fstab* untuk letak partisi *hardisk* yang digunakan. Konfigurasi yang digunakan yaitu:

```
default 0
timeout 0
hiddenmenu
title Ubuntu 14.04.05 LTS, kernel 3.13.0.0-100-generic
root (hd0,0)

kernel /boot/vmlinuz-3.13.0-100-generic
BOOT_IMAGE=/boot/vmlinuz-3.13.0-100-generic root=UUID=93c810bb-
4bbf-437f-8713-cc5b6d3333d5 ro console=tty1 console=ttyS0

initrd /boot/initrd.img-3.13.0-100-generic
```

Keterangan:

`default 0` : menspesifikasikan section pertama dari file akan dieksekusi sebagai default section

`timeout 0` : merupakan batas waktu yang diberikan kepada user untuk melakukan interrupt pada saat startup

`hiddenmenu` : menspesifikasikan bahwa boot menu akan dihidden secara default

`title` : merupakan judul dari grub boot loader

`root (hd0,0)` : merupakan alamat dari root device yang digunakan

`kernel` : merupakan referensi load kernel

`initrd` : merupakan initial ramdisk yang digunakan untuk temporary file root sistem

Sedangkan untuk contoh konfigurasi file `fstab` yang digunakan yaitu

```
/dev/xvda1 / ext4 defaults 1 1
```

Keterangan:

`/dev/xvda1` : merupakan alamat blok device yang digunakan

`/` : merupakan alamat mount directory yang digunakan dalam hal ini adalah root

`ext4` : merupakan tipe file sistem yang digunakan

`defaults` : merupakan opsi associated mount yang digunakan

`1 1` : merupakan order untuk dump dan fsck

Hasil akhir dari proses ekspor ini yaitu file image berbentuk raw kemudian dikompres menjadi format `tarball` yang selanjutnya akan diupload ke server Google Compute Engine.

2. Impor image di server Google Compute Engine

Setelah proses ekspor berhasil yang ditandai dengan terbentuknya *single image file* dalam format *tarball*. Proses selanjutnya yaitu melakukan impor VM image tersebut ke dalam server Google Storage. Google Storage merupakan salah satu layanan yang disediakan oleh Google Cloud dalam bentuk *storage* atau tempat penyimpanan. Setelah file berhasil terupload pada Google Storage, proses selanjutnya yaitu melakukan proses import melalui CLI (Command Line Interface) dengan menggunakan perintah import VM yang telah disediakan oleh Google Compute Engine

Contoh perintah impor yang digunakan yaitu:

```
gcloud compute images create aws-ubuntu --source-url  
gs://fileamazon/tes.tar.gz
```

Keterangan :

`gcloud compute images create` : digunakan untuk melakukan proses impor dari custom image yang telah dibuat

`aws-ubuntu` : merupakan nama VM yang akan dibuat

`-source-url gs://fileamazon/tes.tar.gz` : merupakan alamat dari file image yang akan diimpor

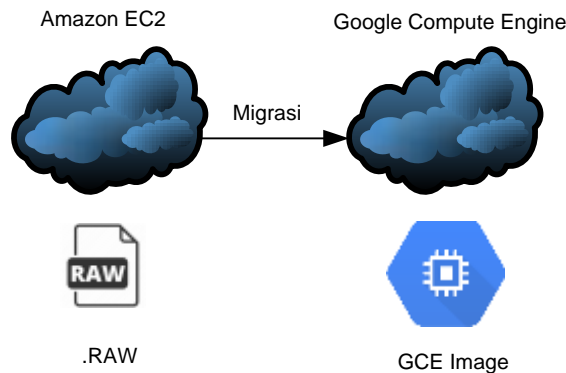
Ketika proses impor berhasil, VM import tersebut akan diubah oleh sistem Google Cloud menjadi *image google*. Pembuatan VM baru dalam *server* Amazon EC2 memakai *image* hasil import server Amazon EC2 tersebut. Saat pembuatan VM baru, pengguna dapat menspesifikasikan ulang konfigurasi *hardware* dan keamanan yang akan digunakan menyesuaikan dengan pilihan yang tersedia dalam Google Cloud Engine. Proses akhir migrasi *image* VM ini diakhiri dengan menghidupkan VM baru yang telah dibangun menggunakan *image* VM dari server Amazon EC2, jika proses *booting* dan *startup* berhasil dan aplikasi di dalam sistem berjalan sesuai dengan kondisi di *server* fisik maka proses migrasi sistem dari *server* Amazon EC2 ke *server* Google Compute Engine telah berhasil.

Tabel 4.3 Hasil Pengujian Migrasi Image VM Amazon EC2 ke Google Compute Engine

No	Jenis Pengujian		Hasil
1	Migrasi VM Amazon EC2 ke Google Compute Engine	Bootting	Berhasil
		Login	Berhasil

Hasil pengujian menunjukkan bahwa untuk seluruh pengujian migrasi VM dari server Amazon EC2 ke Google Cloud Engine telah berhasil ditunjukkan dengan berhasilnya proses booting dan login ke VM setelah proses migrasi. Berdasarkan data hasil pengujian migrasi image VM dari server Amazon EC2 ke server Google Compute Engine proses migrasi yang dilakukan yaitu berhasil.

Proses teknis migrasi untuk skenario kedua ini berbeda dengan skenario pertama. Jika skenario pertama proses migrasi diharuskan dimulai dari *server* fisik karena menggunakan fasilitas VM Export/Import yang disediakan oleh Amazon



Gambar 4.8 Proses Perubahan Image VM dari Amazon EC2 ke Google Compute Engine

Proses perubahan *image* selama migrasi menjadi hal yang penting karena menyangkut integritas data yang terdapat pada *image* tersebut, karena terdapat banyak kemungkinan *image* akan mengalami *corrupt* baik saat proses import/eksport maupun saat *upload* dan *download image* di masing-masing *server*. Format VM image di Amazon EC2 yaitu *.RAW* yang dikompres dalam bentuk *tarball* kemudian ketika dilakukan migrasi ke server Google Compute Engine dilakukan transformasi menjadi format GCE Image dan hasil akhirnya sistem dapat berjalan seperti aslinya sampai dengan berhasil diaksesnya layanan *website* E-Commerce oleh *client* di *server* Google Compute Engine.

Pada saat perpindahan migrasi VM dari Amazon EC2 sampai ke Google Compute Engine, credential login sistem operasi yang dipakai tidak mengalami perubahan selama migrasi. Dalam aspek konfigurasi *security* ketika proses migrasi di Google Compute Engine berhasil, pada saat mengakses/*remote* server di Google Compute Engine harus tetap memakai *public & private key* Amazon. EC2 Selain itu konfigurasi IP pada saat setelah migrasi berubah baik di Amazon EC2 maupun di Indonesian Cloud. Jumlah vCPU, besar *memory* dan *hardisk* ketika migrasi ke Google Compute Engine dapat diubah sesuai keinginan.

Untuk pengujian fungsionalitas aplikasi pada Google Compute Engine seluruh test case fungsionalitas yang diujikan berhasil secara keseluruhan dan tidak ada perubahan fungsionalitas seperti aslinya yang ditunjukkan tidak ada

error/defect software baik mayor maupun minor dari 13 aktifitas pegujian aplikasi yang dilakukan.

Tabel 4.4 Hasil Pengujian Fungsionalitas Aplikasi dari Amazon EC2 ke Google Compute Engine

No	Aktifitas	Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
		Defect Major	Defect Minor	Defect Major	Defect Minor
1	Memesan Barang	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
2	Menambah Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
3	Mengubah Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
4	Menghapus Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
5	Menambah Komentar Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
6	Mensubmit Review Produk	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
7	Menambah Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
8	Mengubah Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
9	Menghapus Artikel	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
10	Melihat Report	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
11	Menambah User Baru	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
12	Mengubah User	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
13	Menghapus User	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

Berdasarkan keseluruhan pengujian yang telah dilakukan baik pengujian migrasi image VM dan pengujian fungsionalitas aplikasi di masing-masing *server* migrasi dapat ditentukan level interoperabilitasnya.

Tabel 4.5 Hasil Evaluasi Secara Keseluruhan

No	Pengujian	Berhasil	Peringatan	Gagal
1	Migrasi Sistem			
1a	Amazon EC2 ke Indonesian Cloud	Berhasil	-	-
1b	Amazon EC2 ke Google Compute Engine	Berhasil	-	-
2	Fungsionalitas Aplikasi			
2a	Amazon EC2 ke Indonesian Cloud	-	Peringatan (Banner website tidak muncul pada homepage aplikasi.)	-
2b	Amazon EC2 ke Google Compute Engine	Berhasil	-	-

Berdasarkan hasil evaluasi interoperabilitas secara keseluruhan, migrasi skenario kedua yaitu dari server Amazon EC2 ke Google Compute Engine mempunyai tingkat interoperabilitas yang lebih unggul dari pada skenario pertama yaitu dari Amazon EC2 ke Indonesian Cloud, yang disebabkan karena adanya kendala munculnya *error* atau *defect software* minor saat selesai migrasi yang menyebabkan level interoperabilitas migrasi dari Amazon EC2 ke Indonesian Cloud digolongkan ke level “Peringatan”. Tetapi jika dibandingkan dengan parameter yang lain seperti kemudahan migrasi, skenario pertama lebih unggul dikarenakan masing-masing cloud menyediakan tools yang memudahkan proses migrasi antar cloud.

Berdasarkan hasil pengujian untuk skenario pertama yaitu migrasi Amazon EC2 ke Indonesian Cloud terdapat 3 *test case* yang gagal dari keseluruhan pengujian dimana total keseluruhan pengujian yang diujikan yaitu berjumlah 97 *test case* sehingga tingkat interoperabilitas dari migrasi Amazon EC2 ke Indonesian Cloud yaitu 99.97%. Sedangkan untuk migrasi dari Amazon EC2 ke Google Compute Engine, tidak ada *test case* yang gagal sehingga tingkat interoperabilitasnya 100%.

Jika dibandingkan dengan penelitian sebelumnya dimana hanya melakukan verifikasi pengujian terhadap sistem operasi pada migrasi antar *hypervisor* yang berbeda parameter yang diukur oleh penelitian sebelumnya lebih banyak daripada parameter yang diuji pada penelitian ini. Parameter VM lifecycle yang diuji oleh penelitian sebelumnya yaitu Launch, Reboot, Pause, Un-Pause, Suspend, Resume, dan Terminate. Sedangkan yang parameter yang diuji oleh penelitian ini yaitu Booting dan Login.

Tabel 4.6 Perbandingan Hasil dengan Penelitian Sebelumnya

Skenario Migrasi	Penelitian Sebelumnya			Penelitian Yang Telah Dikerjakan		
	Hypervisor Asal	Hypervis or Tujuan	Hasil	Hypervisor Asal	Hypervisor Tujuan	Hasil
Migrasi dari Amazon EC2 ke Indonesian Cloud						
Migrasi Server Fisik ke Amazon EC2	VMWare	Xen	Warning	VMWare	Xen	Sukses
Migrasi Amazon EC2 ke Indonesian Cloud	Xen	VMWare	Failure	Xen	VMWare	Sukses
Migrasi dari Amazon EC2 ke Google Compute Engine						
Migrasi Amazon EC2 ke Google Compute Engine	Xen	KVM	Failure	Xen	KVM	Sukses

Berikut perbandingan teknologi yang digunakan oleh masing-masing provider dalam proses *migrasi image* Virtual Machine

Tabel 4.7 Perbedaan Teknologi Migrasi Antar Provider

Jenis Teknologi yang digunakan	Server Fisik	Amazon EC2	Indonesian Cloud	Google Compute Engine
Migrasi dari Amazon EC2 ke Indonesian Cloud				
Type Hypervisor	VMWare ESXi	XEN	VMWare vCloud Air	
Type Virtualisasi	Tipe 1 / Bare Metal	Bare Metal HVM (hardware virtual machine)	Tipe 1 / Bare Metal	
Format Image	OVA	AMI	OVA	
Storage Penyimpanan Image Setelah Import/Eksport	Tidak ada	Amazon S3	Tidak ada	
Tools untuk Proses Eksport/ Import	Proses eksport menggunakan VMWare Vsphere Desktop Client	Proses import menggunakan CLI melalui perintah <code>aws ec2 import-image</code> Proses eksport menggunakan	Proses import menggunakan tools vCloud Director	

Jenis Teknologi yang digunakan	Server Fisik	Amazon EC2	Indonesian Cloud	Google Compute Engine
		CLI melalui perintah <code>ec2-create-instance-export-task</code>		
Migrasi dari Amazon EC2 ke Google Compute Engine				
Tipe Hypervisor		XEN		KVM
Tipe Virtualisasi		Bare Metal HVM (hardware virtual machine)		Linux Bare Metal
Format Image		AMI		RAW
Storage Penyimpanan Image Setelah Import/Eksport		Tidak ada		Google Cloud Storage
Tools untuk Proses Eksport/Import		Proses eksport menggunakan CLI melalui perintah <code>ec2-bundle-vol</code>		Proses import menggunakan CLI melalui perintah <code>gcloud compute images</code>

Halaman ini sengaja dikosongkan

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Proses migrasi sistem antar *cloud provider* yang berbeda dengan 2 skenario, pertama dari Amazon EC2 ke Indonesian Cloud kemudian dari Amazon EC2 ke Google Compute Engine telah berhasil dilakukan.
2. Berdasarkan hasil dari pengujian fungsionalitas aplikasi setelah migrasi dari Amazon EC2 ke Indonesian Cloud yang telah dilakukan, terdapat error/defect software yang terjadi setelah migrasi, walaupun masih termasuk kategori minor yang menyebabkan level interoperabilitas migrasi dari Amazon EC2 ke Indonesian Cloud digolongkan menjadi level “Peringatan”.
3. Sedangkan berdasarkan hasil pengujian fungsional aplikasi setelah migrasi dari Amazon EC2 ke Google Compute Engine seluruh test menunjukkan tidak ada *error/defect software* yang terjadi sehingga dapat digolongkan tingkat interoperabilitas aplikasi saat migrasi dari Amazon EC2 ke Google Compute Engine masuk dalam level “Berhasil”.
4. Tingkat interoperabilitas migrasi dari Amazon EC2 ke Google Compute Engine lebih unggul dari pada Amazon EC2 ke Indonesian Cloud.
5. Berdasarkan kemudahan dari sisi teknis migrasi, migrasi dari Amazon EC2 ke Indonesian Cloud lebih mudah dikarekan tersedianya tools yang memudahkan pengguna untuk melakukan proses migrasi.
6. Berhasilnya proses migrasi Virtual Machine antar cloud tidak menjamin bahwa fungsionalitas aplikasi yang berada dalam sistem tersebut berjalan dengan baik, untuk itu diperlukan pengujian yang lebih dalam mengenai fungsionalitas aplikasi yang berada dalam sistem tersebut.

5.2 Saran

Untuk penelitian selanjutnya diharapkan untuk dapat menambah tipe pengujian aplikasi, bukan hanya dari segi fungsionalitas saja, tetapi juga dari segi pembangunan kode, kompatibilitas, performansi, beban, regresi, integritas data, serta dari segi keamanan. Selain itu untuk penelitian selanjutnya juga disarankan untuk memperbanyak variasi *cloud provider* serta tipe *hypervisor* yang digunakan sehingga dapat dihasilkan hasil pengujian yang lebih komprehensif.

DAFTAR PUSTAKA

- Amazon Web Services (2016), *Amazon EC2*, Entry From : <https://aws.amazon.com/ec2>
- Amazon Web Services (2016). *VM Import/Export User Guide: Troubleshooting VM Import/Export*. Entry From: <http://docs.aws.amazon.com/vm-import/latest/userguide/vmimport-troubleshooting.html>
- Badola, V(2015). *AWS AMI Virtualization Types: HVM vs PV (Paravirtual VS Hardware VM)*. Entry From: <http://cloudacademy.com/blog/aws-ami-hvm-vs-pv-paravirtual-amazon/>
- Beda, J. (2012), *Google Compute Engine: A Technical Intro Velocity Europe 2012.*, Entry From : <http://slides.eightypercent.net/velocity-europe-2012>
- Cai, J., dan Hu, Q. (2011) ,“ Analysis for Cloud Testing of Web Application”, *Proceedings of 2012 2th International Conference on System and Informatics*, Shanghai, hal. 293-297.
- Cloud Standards Customer Council (2013), *Migrating Applications to Public Cloud Services: Roadmap for Success*,
- Dowell, S., Barreto, A., Michael, J.B., dan Shing, M. (2011) “Cloud to Cloud Interoperability”, *Proceedings of 2011 6th International Conference on System of Systems Engineering*, Albuquerque, hal. 258-263.
- Galin, D.(2004),*Software Quality Assurance From theory to implementation*, Pearson Education Limited, Essex
- Gholami, M.F., Daneshgar, F., Low, G., dan Beydoun, G (2016). *Cloud migration process—A survey, evaluation framework, and open challenges*, The Journal of Systems and Software 120, hal 31-69
- Indonesian Cloud(2016), *Cloud Server*, Entry From : <https://indonesiancloud.com/id/hosting/cloud-server/>
- Jadeja, Y., and Modi, K. (2012), “Cloud Computing - Concepts, Architecture and Challenges” *Proceedings of 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET]*, Kumaracoil, hal. 877-880.
- Lenk, A., Katsaros, G., Menzel, M., Skipp, R., Rake-Relevant, J., Castro-Leon, E., dan V P, Gopan. (2014), “TIOSA: Testing VM Interoperability at an

- OS and Application Level”, *Proceedings of IEEE International Conference on Cloud Engineering*, Boston MA, hal. 245-252.
- Lewis, G.A. (2013) “Role of Standards in Cloud-Computing Interoperability”, *Proceedings of IEEE 46th Hawaii International Conference on System Sciences*, Wailea, hal. 1652-1661.
- Open Data Center Alliance (2013), *Implementing the Open Data Center Alliance Virtual Machine Interoperability Usage Model*. Entry From : <http://www.opendatacenteralliance.org>
- Open Data Center Alliance (2013), *Virtual Machine (VM) Interoperability in a Hybrid Cloud Environment Rev. 1.2*. Entry From : <http://www.opendatacenteralliance.org>
- Opera-Martin, J., Justice, S. Reza, dan Tian, F. (2014), “Critical Review of Vendor Lock-in and Its Impact on Adoption of Cloud Computing”, *Proceedings of 2014 International Conference on Information Society (i-Society 2014)*, London, hal. 92-97.
- Pappas, A.W. (2014), *Migration of Legacy Applications to the Cloud: A Review on Methodology and Tools for Migration to the Cloud*, Tesis Ph.D, Umea Universitet, Sweden
- PCI Security Standard Council (2011), *Information Supplement: PCI DSS Virtualization Guidelines*, Entry From : https://www.pcisecuritystandards.org/documents/Virtualization_InfoSupp_v2.pdf
- Srikanth, H., Cashman, M., dan Cohen, M.B (2016), *Test case prioritization of build acceptance tests for an enterprise cloud application: An industrial case study*, *The Journal of Systems and Software* 119, hal 122-135
- Sparksupport (2013), *Server Virtualization*, Entry from SparkSupport Infotech Pvt Ltd. info@sparksupport.com
- Sosinky, B. (2011), *Cloud Computing Bible*, Wiley Publishing Inc., Indianapolis.
- Strunk, A., dan Dargie, W. (2013) “Does Live Migration of Virtual Machines cost Energy?”, *Proceedings of 2013 IEEE 27th International Conference on Advanced Information Networking and Applications*, Barcelona, hal. 514-521.
- Villary, M., Brandic, I., Tusa, F. (2012), *Achieving Federated and Self-Manageable Cloud Infrastructure: Theory and Practice*, Business Science Reference(IGI Global), Hershey PA

VMWare (2015), *vSphere 5.1 Documentation Center: Introduction to VMware vSphere Virtual Machines* . Entry From : <https://www.pubs.vmware.com>

VMWare(2015), Datasheet VMware vCloud Air Service Description, <http://www.vmware.com/files/au/pdf/vcloud-air/vcloud-air-Datasheet.pdf>

Halaman ini sengaja dikosongkan

LAMPIRAN

Hasil Pengujian Fungsionalitas Aplikasi

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
1	Memesan Barang	1.1 Mengakses Halaman Homepage	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Banner Website Tidak Muncul	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.2 Memilih Produk	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.3 Masuk ke Keranjang Belanja	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.4 Memasukkan Kupon	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.5 Mengubah Keranjang Belanja	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.6 Melakukan Proses Checkout	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		1.7 Login	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.8 Mengisi Form Registrasi Akun	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.9 Mengisi Form Alamat Pengiriman	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.10 Memilih Metode Pembayaran	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.11 Memesan Barang	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.12 Mengurangi Stok Barang	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.13 Memproses Pemesanan	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		1.14 Melihat Status Pemesanan	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
2	Menambah Produk	2.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		2.2 Mengakses Halaman Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.3 Mengakses Halaman Add Product	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.4 Mengisi Deskripsi Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.5 Mengunggah Gambar Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.6 Mempublish Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.7 Menampilkan Produk	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.8 Melihat Produk Baru	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		2.9 Melihat Produk Baru	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
3	Mengubah Produk	3.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		3.2 Mengakses Halaman Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		3.3 Mengakses Halaman Edit Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		3.4 Mengubah Informasi Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		3.5 Mengupdate Informasi Produk Terbaru	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		3.6 Melihat Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		3.7 Melihat Produk	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
4	Menghapus Produk	4.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		4.2 Mengakses Halaman Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		4.3 Memindah Produk Ke Menu Trash	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		4.4 Mengakses Halaman Trash	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		4.5 Menghapus Permanen Produk	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		4.6 Menghapus Permanen Produk	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
5	Menambah Komentar Artikel	5.1 Mengakses Halaman Homepage	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Banner Website Tidak Muncul	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.2 Mengakses Halaman Artikel	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.3 Login	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.4 Mengisi form komentar	Customer	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
				Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada
		5.5 Submit komentar	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.6 Menyetujui komentar	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.7 Menampilkan komentar	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.8 Melihat komentar	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		5.9 Melihat Komentar	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
6	Mensubmit Review Produk								Banner Website Tidak Muncul				
		6.1 Mengakses Homepage	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada		Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.2 Mengakses Detail Produk	Customer	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
				Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada
		6.3 Login	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.4 Mengisi Kolom Review	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.5 Submit Review	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.6 Menyetujui Review	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.7 Menampilkan Review Customer	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.8 Melihat Review Customer	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		6.9 Melihat Review	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
7	Menambah Artikel	7.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		7.2 Mengakses Halaman	Administrator	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak	Tidak

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		Artikel		Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada
		7.3 Mengakses Halaman Tambah Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		7.4 Mengisi Informasi Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		7.5 Publish Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		7.6 Menampilkan Artikel Baru	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		7.7 Melihat Artikel Baru	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		7.8 Melihat Artikel Baru	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
8	Mengubah Artikel	8.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		8.2 Mengakses Halaman Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		8.3 Mengakses Halaman Ubah Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		8.4 Mengubah Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		8.5 Menampilkan Artikel Terbaru	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		8.6 Melihat Artikel Terbaru	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		8.7 Melihat Artikel Terbaru	Customer	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
9	Menghapus Artikel	9.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		9.2 Mengakses Halaman Artikel	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		9.3 Memilih Menu Trash	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		9.4 Mengakses Halaman Trash	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		9.5 Memilih Menu Hapus Permanen	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		9.6 Menghapus Artikel	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
				Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada
10	Melihat Report	10.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		10.2 Mengakses Halaman Report	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		10.3 Memilih Tipe Report	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		10.4 Download File Laporan	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
11	Menambah User Baru	11.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		11.2 Mengakses Halaman User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		11.3 Mengakses Halaman Tambah User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		11.4 Mengisi Informasi User Baru	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		11.5 Menyimpan Data User Baru	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		11.6 Menampilkan Data User Baru	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
12	Mengubah User	12.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		12.2 Mengakses Halaman User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		12.3 Mengakses Halaman Ubah User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		12.4 Mengubah Data User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		12.5 Menyimpan Perubahan User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		12.6 Menampilkan Data User Terbaru	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
13	Menghapus User	13.1 Login	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		13.2 Mengakses Halaman User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		13.3 Memilih Menu Hapus	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

No	Aktifitas	Detail Aktifitas	Aktor	Migrasi Amazon EC2 ke Indonesian Cloud						Migrasi Amazon EC2 ke Google Compute Engine			
				Hasil Pengujian Server Fisik		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Indonesian Cloud		Hasil Pengujian Server Amazon EC2		Hasil Pengujian Server Google Compute Engine	
				Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor	Defect Major	Defect Minor
		User		Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada	Ada
		13.4 Konfirmasi Penghapusan User	Administrator	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada
		13.5 Menghapus Data User	Sistem	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada	Tidak Ada

BIOGRAFI PENELITI



Soffa Zahara, lahir di Ngawi, 4 Juli 1991. Merupakan lulusan D3 Jurusan Teknik Komputer Telkom University, dan S1 Teknik Informatika Universitas Pasundan Bandung. Peneliti merupakan dosen jurusan Teknik Informatika di Universitas Islam Majapahit Mojokerto. Peneliti dapat dihubungi melalui email soffa.zahara@gmail.com dan no telepon 085722423159